NASA
Technical
Paper
2857

November 1988

Development and Flight Test Experiences With a Flight-Crucial Digital Control System

Dale A. Mackall

NASA
Technical
Paper
2857

1988

Development and Flight Test Experiences With a Flight-Crucial Digital Control System

Dale A. Mackall
Ames Research Center
Dryden Flight Research Facility
Edwards, California

NASA
National Aeronautics
and Space Administration

**Scientific and Technical
Information Division**

CONTENTS

Page

SUMMARY 1

1 INTRODUCTION 1

2 NOMENCLATURE 2

3 SYSTEM SPECIFICATION 5
3.1 Control Laws and Handling Qualities 5
3.2 Reliability and Fault Tolerance 5

4 DESIGN 6
4.1 System Architecture and Fault Tolerance 6
4.1.1 Digital flight control system architecture 6
4.1.2 Digital flight control system computer hardware 8
4.1.3 Avionics interface 8
4.1.4 Pilot interface 9
4.1.5 Actuator interface 10
4.1.6 Electrical system interface 11
4.1.7 Selector monitor and failure manager 12
4.1.8 Built-in test and memory mode 14
4.2 Control Laws 15
4.2.1 Control law development process 15
4.2.2 Control law design 15
4.3 Digital Flight Control System Software 17
4.3.1 Software development process 18
4.3.2 Software design 19

5 SYSTEM-SOFTWARE QUALIFICATION AND DESIGN ITERATIONS 19
5.1 Schedule 20
5.2 Software Verification 21
5.2.1 Verification test plan 21
5.2.2 Verification support equipment 22
5.2.3 Verification tests 22
5.2.4 Reverifying the design iterations 24
5.3 System Validation 24
5.3.1 Validation test plan 24
5.3.2 Support equipment 25
5.3.3 Validation tests 25
5.3.4 Revalidation of designs 33
5.4 Qualification Issues 33

6 CONFIGURATION CONTROL 33

7 FLIGHT TEST 34
7.1 General 35
7.2 Fault-Tolerant Design 37
7.2.1 In-flight experience 37
7.2.2 Ground experience 39
7.2.3 Summary 40
7.3 Control Laws 41

7.4 Hardware 42
7.5 Software 43

8 OBSERVATIONS AND RECOMMENDATIONS 44
8.1 Anomaly of Flight 44, A Case Study 44
8.1.1 Specifcation 44
8.1.2 Design 45
8.1.3 Qualification 45
8.2 Observations and Recommendations by development Phase 45
8.2.1 Specification 45
8.2.2 Design 46
8.2.3 Qualification 47
8.2.4 Flight test 48

9 CONCLUDING REMARKS 49

APPENDIX – CONTROL LAW VERIFICATION REQUIREMENTS 50

REFERENCES 52
TABLES 53
FIGURES 81

SUMMARY

Engineers and scientists in the advanced fighter technology integration (AFTI) F-16 program investigated the integration of emerging technologies into an advanced fighter aircraft. AFTI'S three major technologies included (1) flight-crucial digital control, (2) decoupled aircraft flight control, and (3) integration of avionics, flight control, and pilot displays. In addition to investigating improvements in fighter performance, researchers studied the generic problems confronting the designers of highly integrated flight-crucial digital control systems.

The author provides an overview of both the advantages and problems of integrated digital control systems. An examination of the specification, design, qualification, and flight test life-cycle phase is provided. An overview is given of the fault-tolerant design, multimoded decoupled flight control laws, and integrated avionics design. The approach to qualifying the software and system designs is discussed, and the effects of design choices on system qualification are highlighted.

AFTI F-16 flight test results are summarized for the fault-tolerant, decoupled flight control, hardware, and software requirements. The effects of design choices and qualification procedures on flight test operations are detailed, based on AFTI flight experience.

Observations and recommendations are given for each development phase — specification, design, qualification, and flight test.

1 INTRODUCTION

The advanced fighter technology integration (AFTI) F-16 program provided the opportunity to investigate the benefits and complexities of integrating advanced aircraft technologies into a fighter aircraft. The study was a joint National Aeronautics and Space Administration (NASA), U.S. Air Force, and U.S. Navy program and was managed by the Air Force Flight Dynamics Laboratory. NASA goals were to ensure safety during flight testing and to provide an independent assessment of the advanced technologies.

The primary subject of this report is the digital flight control system (DFCS) and its integration with the avionics and pilot displays. An introduction to the history, rationale, and nomenclature of digital flight control systems can be found in Szalai (1978). The AFTI F-16 DFCS development objectives included assessment of a triplex dual-fail operate architecture, integration of avionics and pilot displays with the DFCS, and development of mission-specific decoupled flight control modes.

Operating a DFCS without mission impairment after any two failures required a minimum of four channels of redundancy in previously designed systems. If a triplex system could correctly choose between the remaining two channels when the second failure occurred, acquisition and maintenance costs for the flight control system could be reduced. Reducing pilot workload and increasing weapon effectiveness were the goals of integrating the DFCS and its mission-specific decoupled control modes with the avionics system and pilot displays. In previously designed systems, the flight controls did not have specific modes for the different missions. The pilot was required to individually configure each avionic system for a mission.

This report includes an historical review of the development and flight test of this integrated DFCS program. The historical review is structured to provide an adequate background of the development process and the resulting design needed to comprehend the flight test results. The author addresses each of the development phases — specification, design, qualification, and flight test. Important lessons learned are illustrated with examples from flight test experience.

The increasing use of system integration to increase aircraft performance, and the flight crucial nature of these systems, dictates a thorough assessment of this integrated DFCS program.

2 NOMENCLATURE

AAG	air-to-air gunnery
ACK	acknowledge
A-D	analog to digital
ADI	attitude directional indicator
AFTI	advanced fighter technology integration
AGL	above ground level, ft
AIU	actuator interface unit
ALT	altimeter
AMUX	avionics multiplex bus
A/S	airspeed
ASB	air-to-surface bombing
ASG	air-to-surface gunnery
ATP	acceptance test procedure
A_y	lateral acceleration, ft/sec^2
ac	alternating current
alpha	angle of attack, deg
a_n	normal acceleration, ft/sec^2
BIT	built-in test
beta	angle of sideslip, deg
CADC	central air data computer
CCV	control configured vehicle
CHGR	charger, battery
CPC	computer program component
CPDS	computer program development specification
CPPS	computer program product specification
CPU	central processing unit
c.g.	center of gravity, percentage mean aerodynamic chord
D-A	digital to analog
DAAG	decoupled air-to-air gunnery
DASB	decoupled air-to-surface bombing
DASG	decoupled air-to-surface gunnery
DFCS	digital flight control system

DGFT	dog fight
DN	down
DNRM	decoupled normal
DST	device status table
dc	direct current
deg	degrees
deg/sec	degrees per second
EMIC	electromagnetic interference and compatibility
EPU	emergency power unit
ETSE	engineering test support equipment
FCC	fire control computers
FCR	fire control radar
FDIR	fault detection, indentification, and reconfiguration
FLCC	flight control computer
FM	failure manager, a software component
FMET	failure modes and effects testing
FPME	flightpath maneuver enhancement
flt	flight
ft	feet
GCA	good channel average
GCMD	G command
g, G	longitudinal acceleration, g
HSI	horizontal situation indicator
HUD	head-up display
Hz	hertz
hr	hours
IBU	independent back-up unit
IFFC	integrated flight fire control
ILS	instrument landing system
INU	inertial navigation unit
IOC	input-output controller
ISA	integrated servoactuator
KCAS	knots calibrated airspeed
k	thousand
LARAP	low-altitude radar autopilot
LAT-DIR	lateral-directional
LCND	left canard
LEF	leading-edge flap
LFLP	left trailing edge flap
LH	left hand
LHT	left horizontal tail
LOC	location in memory
LQS	linear quadratic synthesis
LRU	line replaceable unit
lb	pounds
M	Mach
MAX	maximum afterburner power

MHz	million hertz
MIL	military power
MPD	multipurpose display
MSL	median select logic
MSOV	missile override
msec	millisecond
N_X	longitudinal load factor, g
N_Y	lateral load factor, g
N_Z	normal load factor, g
OFP	operational flight program
P	roll rate, deg/sec
$\dot{P}$	roll acceleration, deg/sec^2
PDG	programmable display generator
PLA	power lever angle, deg
PMG	permanent magnet generator
PRME	pitch rate maneuver enhancement
PS	pressure system
PSA	pneumatic sensor assembly
P_S	static pressure
lb/ft^2	pounds per square foot
Q	pitch rate, deg/sec
Q_C	impact pressure
R	yaw rate, deg/sec
RAM	random-access memory
RCND	right canard
RFLP	right flap
RH	right hand
RHT	right horizontal tail
RM	redundance managment
ROM	read only memory
RUD	rudder
rad	radian
rad/sec	radians per second
recon	reconfiguration
rpm	revolutions per minute
SAAG	standard air-to-air gunnery
SASB	standard air-to-surface bombing
SASG	standard air-to-surface gunnery
S/M	selector monitor
SMS	stores management set
S/N	serial number
SNRM	standard normal mode
SOW	statement of work
SP	pitch stick
SR	roll stick
SV1,2,3	servovalues 1,2,3
sec	second
T	throttle twist

TCO	total computed output
TEF	trailing edge flaps, deg
TR	transformer rectifier
V	velocity vector
V and V	verification and validation
VA	volt-amps
V ac	volts, ac
VCRI	verification cross reference index
V dc	volts, dc
VID	video
ω	frequency, rad/sec

3 SYSTEM SPECIFICATION

In the system specification phase, operational requirements are detailed to a level the designers can use.

The first step in specifying the AFTI F-16 (fig. 1) system design was the statement of work (SOW) released on November 16, 1978 by the Air Force Wright Aeronautical Laboratories, Wright Patterson Air Force Base, Ohio. This document specified the requirements for decoupled control, weapon line pointing, aerodynamic vehicle modifications, digital flight control system, and pilot-vehicle interface. These general requirements were then detailed in the following categories: air vehicle, systems engineering, test and evaluation. The contractor, General Dynamics, in Fort Worth, Texas, was responsible for the second step in specifying the system. After the contractor generated the system specification, an entire tree of specifications grew for each system, new or modified, that was required to accomplish AFTI objectives. The following paragraphs in section 3 will address the specifications only as they apply to the digital flight control system.

3.1 Control Laws and Handling Qualities

The SOW specified the requirements for the unique decoupled control modes (table 1) and the airframe stability and flying qualities requirements. Decoupled control requirements included direct lift and sideforce, fuselage pointing independent of flight path, vertical and lateral translation, and wings level steering. The stability and flying qualities requirements were based on MIL-F-8785C (U.S. Department of Defense, 1980). From the SOW, the contractor provided the detailed requirements for aircraft stability and flying qualities. Requirements included short-period damping ratio limits, short-period frequency requirements, dutch roll frequency and damping, and force gradient limits for controllers.

3.2 Reliability and Fault Tolerance

The reliability and fault tolerance requirements from the SOW are shown in table 2. These include reliability, fail-operational, switching and failure transients, and cooling requirements. A requirement was a 95-percent chance of being fully operational for a second failure of a similar device. MIL-F-9490D (U.S. Department of Defense, 1975) provided the requirements for DFCS development.

Software requirements stated that the contractor develop, validate, and maintain the software in accordance with a software development and management plan prepared by the contractor. It identified the procedures and methodologies for verification and validation, documentation, and control of software. The requirement for an independent backup unit (IBU) for the DFCS was identified. The IBU provided an analog backup to the primary DFCS that is independent of the DFCS software. Level 3 flying qualities (U.S. Department of Defense, 1980) throughout the flight test envelope and level 2 flying qualities in landing were specified for the IBU.

The electrical system was required to provide power to support DFCS reliability requirements. System level and DFCS specifications from the contractor restated the requirements of the SOW, identified quality assurance provisions, and provided a comprehensive design criterion for the DFCS and its components, redundancy levels, and their fail-operational capabilities (table 3).

The quality assurance section of the specifications provides a table that cross-referenced system requirements to verification methods (table 4). The reliability aspects are shown to be verified through analysis only (items 3.2.3.1 and 3.2.3.2 of table 4). In nonredundant systems that consist of hardware only, analysis techniques, such as fault trees, are sufficient. However, in redundant, software-driven systems, ground test and demonstrations are also needed to verify reliability. Hence, extensive failure modes and effects testing were developed (section 5).

Documents that specify the software development are identified in the specifications by title only. All relevant military standards are identified.

4 DESIGN

This section contains the DFCS design and provides an overview of the methods used to obtain it. The design issues addressed are (1) system architecture and fault tolerance aspects, (2) control laws, and (3) software.

4.1 System Architecture and Fault Tolerance

System architecture and fault tolerance are closely associated. The multichannel architecture is a direct result of the need for fault tolerance. Because a large portion of the fault-tolerant design is in software, the software aspects of the fault-tolerant design are also covered in section 4.3. Additional information can be found in Yousey and others (1984).

4.1.1 Digital Flight Control System Architecture

The requirements for the DFCS architecture (fig. 2) were derived directly from the SOW and system specification. This derivation consisted of identifying specific design requirements for each numbered item in the SOW and system specification. For each design requirement, hardware and software resources were then allocated to ensure that the design requirements were met.

For example, the design requirement for six-degree-of-freedom control was achieved using the standard F-16 sensors, the triplex computer set, and the standard F-16 control surfaces plus the canards. Reliability requirements were satisfied by having computer mean-time-between-failure rates and redundancy consistent with those needed to meet probability requirements. Figure 3 shows the reliability of a single-channel design and a triplex-channel design. The failure probability for the triplex-channel system of 1×10^{-10} per flight hr includes the IBU. The major causes of loss of control are discussed in Price and others (1984). Concerns for software reliability were addressed with the inclusion of the IBU. Figure 4 shows the IBU and its relation to the primary DFCS. The IBU can be engaged either manually by the pilot or automatically if proper operation is lost by the three digital processors.

A significant architectural aspect of the DFCS was that the operations of the three computers were not synchronized. This choice was made by the contractor because computer syncronization was believed to introduce a single-point failure caused by electromagnetic interference (EMI) and lightning effects.

To obtain the detailed DFCS architecture, engineering studies and reliability analysis of hardware components were performed; no formal or structured tool was used. Architectural design issues included (1) the design of the IBU and (2) the analog sensor interface.

The IBU trade study addressed (1) how reliable the IBU should be, what redundancy level was needed, and if output command voting would be required; (2) what flight control performance was required of the IBU (requirements were for level 3 handling qualities throughout the flight envelope except for level 2 at landing); (3) what the engagement method should be for the IBU; and (4) how to minimize transients on engagement and disengagement of the IBU. These issues were further complicated by the disagreement between the procuring and flight test agencies regarding flight test of the IBU. The flight test agencies' position to flight test the IBU prevailed and this directly influenced the performance issue and the need for manual IBU engagement and disengagement by the pilot.

A triple redundancy level was chosen for the IBU with a portion of one flight computer card in each of the three DFCS boxes dedicated to the IBU. An output selector, which can select valid commands after a single failure, was included for the horizontal-tail commands to improve the system's fault tolerance in that axis. Space limitations in the computer prohibited output selectors for all surface commands. The performance issue was of constant interest, and the modifications to improve IBU performance continued into flight test. The design of the IBU can be found in Price and others (1984) and in section 4. The IBU modifications included tuning the pitch rate path and providing differential horizontal-tail commands when in manual pitch override (stalls).

The IBU was engaged either manually or automatically. IBU tracking of the primary system for engagement purposes was rejected owing to the need for independence, because a failure in the primary system could not be allowed to affect the IBU's operation. However, the digital system did track the IBU to minimize reengagement transients to the digital system. This was easily accomplished since the digital system monitored the IBU surface commands for inflight failure detection and built-in test (BIT) purposes.

Issues addressed for the sensor interface included (1) the use of digital rather than analog cross-strapping of information (fig. 5), and (2) required sensor sampling rates to minimize differences introduced by the asynchronous computer operation. Analog cross-strapping was first thought to be required to meet data latency and reliability requirements. However, as detailed analysis showed this was not true, digital cross-strapping was chosen because it required less wiring. Digital cross-strapping was accomplished using two dedicated serial transmission lines for each computer. To minimize differences introduced by asynchronous operation, sensors were sampled at four times the basic flight control rate of 64 Hz. This was of particular concern for pilot inputs from the force stick, which can have higher input rates than the aircraft sensors. An assumed worst case input to maximum command was analyzed at 100 percent in 0.1 sec, or 1000 percent/sec (fig. 6). The increased sampling rate reduced the interchannel differences to less than 4 percent for a prefilter break frequency ω of 50 rad/sec. This analysis was also the first to recognize the effect of asynchronous sampling errors. The sampling errors introduced differences between computer channels for each computer-calculated surface command.

4.1.2 Digital Flight Control System Computer Hardware

The flight control computers (FLCC) used were the Bendix (The Bendix Corporation, Teterboro, New Jersey) BDX-930 computers (fig. 7). The basic computer included a central processing unit (CPU), based on a 16-bit, bit-sliced microprocessor and solid-state memory (6K words of random-access memory (RAM) and 24K words of programmable read-only memory). The CPU did not have floating-point capability and was programmed in assembly language. The CPU was supplemented with an input-output controller that performed all input and output data conversion with a single command from the processor. This allowed the processor to compute flight control algorithms without being burdened by inputting and outputting discrete and analog signals.

Additional functions in the flight control computers included a MIL-STD-1553B multiplex data bus interface, failure logic, the IBU, and serial data links to and from each of the other two computers. Failure logic was special circuitry that allowed two computers to fail another; this logic was required to provide the dual-fail-operate capability. A description of the analog and discrete inputs and outputs is provided in table 5. The FLCC represented a state-of-the-art computer in terms of technology used, throughput, and memory.

4.1.3 Avionics Interface

The items that determined the DFCS interface with the avionics were integration of the pilot station with the DFCS to reduce pilot workload, instrumentation of the DFCS, and the use of information from other avionic subsystems. The primary avionics systems (fig. 8) included a fire control computer (FCC), stores management set (SMS), inertial navigation unit (INU), fire control radar (FCR), central air data computer (CADC), two multipurpose displays (MPD), instrumentation system, and a head-up display (HUD). The avionics were interfaced through a MIL-STD-1553B data bus controlled by the fire control computer.

The types of avionics information involved in the DFCS interface and the avionics systems that pass the information are shown in table 6. Pilot mode selection and status information represented the most safety critical data of the avionics interface. Details on the pilot-vehicle interface are given in subsection 4.1.4. Parameters were supplied to the DFCS from the INU, including roll attitude, pitch attitude, and velocity. The parameters were used in the decoupled control modes to assist the pilot during rolling maneuvers.

The ability to instrument and monitor internal DFCS parameters was essential for thorough testing, both in the laboratory and during flight test. The design approach was to have the FCC send the DFCS a list of internal DFCS memory locations to be sent to instrumentation; this list was sent following the running of a BIT. The DFCS did not store its own list because the FCC used nonvolitive core memory and changes to the list could be made more easily. However, this proved not to be true and during flight test, the design was changed to have the DFCS store its own parameter lists. The DFCS could output 64 parameters at a 50-Hz rate.

The MIL-STD-1553B data bus is a dual redundant (one active and one backup) 1-MHz serial bus controlled by the fire control computer. The 1553B data bus performs parity checks and polling tests when transmitting to the other avionics systems. Failure of these tests causes the bus controller to retry the information exchange on the backup bus. If the FCC failed, the stores management set took over bus control.

4.1.4 Pilot Interface

The requirements to integrate the aircraft electronic systems and to reduce pilot workload were accomplished, in part, by the design of the pilot's interface. The concept was to implement mode selection of all aircraft systems — flight control, avionics, and weapons — through a single cockpit action. This concept allowed configuring all aircraft systems for a given mission, such as air-to-air attack, through a single pilot action. Displays of system status were integrated through the use of the multipurpose display (MPD). Flight control, weapons, and radar information could all be displayed on the MPDs. The following discussions in section 4.1.4 only address the interface of the pilot with the DFCS.

The pilot's interface to the DFCS consists of pilot's controllers, discrete switches, dedicated warning lights, two MPDs, and a HUD (fig. 9). The pilot's interface provided aircraft control, DFCS status, mode selection, and failure resets.

The pilot's controllers consisted of a right-hand force stick for pitch and roll control, rudder pedals for directional control, and a throttle grip which twists for decoupled pitch control. The right-hand controller is shown in figure 10. Each controller had a different command characteristic, depending on the active control mode. The relationship of control modes to controller commands is shown in figure 11. Note how the decoupled motion obtained by the rudder pedal and twist grip command changes for different modes. Descriptions of the decoupled control options are shown in figure 12.

Discrete switches in the cockpit were kept to a minimum. They included aircraft trim, decoupled mode selection, a normal acceleration limit engagement switch, IBU switch, and failure resets. Several switches used by other aircraft systems, but related to the flight control system, include speed brake switch and throttle at military and idle positions.

Flight control modes were selected in several different ways. Decoupled mode and IBU were selected using switches on the right-hand control stick. Selection of the different mission control modes — air to air and air to ground — were made in conjunction with avionic and weapon system changes through the HUD panel (fig. 9) or a switch on the throttle (air to air only). Selection of mission specific flight control modes, independent of the other aircraft systems, could be made through the MPDs. In all cases mission specific control modes were selected through the SMS. The SMS sends control mode requests to the DFCS over the 1553B multiplex bus. The mode selection data flow is summarized in figure 13.

The status of the DFCS was presented to the pilot in three ways — warning lights, MPD messages, and HUD displays. The dedicated failure lights warned the pilot of failures detected by the DFCS; the pilot would then use the MPDs to determine the exact failures detected. The HUD information was primarily related to control of the aircraft, but also provided some information on failure aspects of the DFCS.

The MPDs were designed to be the primary interface between the pilot and the DFCS. The MPD functions, HUD indications, and failure lights are listed and briefly described in table 7. Figure 14 shows the DFCS base page and the selection of the DFCS test page.

There were two major concerns with the pilot interface to the DFCS. The first concern was the lack of redundancy in the command path for mission-specific control mode selection. This concern for redundancy in the pilot's control mode selection proved to be valid, as an in-flight anomaly showed during flight test (see section 7.2.1). The second concern was for the method used to display pilot fault information. The fault display complexity is best demonstrated with tables from the pilot's manual and an example. A list of the levels, types, and classes of fault nmemonics for the DFSC is given in table 8, and the fault nmemonics for each of the three categories are described in table 9. Armed with this information, and the ability to decode the hexidecimal numbers into binary numbers, the pilot could determine from a fault display (fig. 15) what the DFCS had declared failed. This fault display tells the pilot that a 1st failure has occurred to an input used in the pitch axis control of the aircraft. Below the English description of the fault, is a two-digit number and three single digits (see fig. 15). The first number can be decoded using table 10, indicating that an angle-of-attack sensor had failed. Table 10 also gives the failure words displayed for each of the categories shown in table 8. The three digits (fig. 15) identify what each computer believes is the failed channel. Each number represents a computer channel — A, B, and C — left to right. A two implies that the channel represented in that column has failed; a four implies that the channel to the left has failed; and a one implies that the channel to the right has failed. In this case the 1st channel, termed channel A, has the failed angle-of-attack sensor. Although some of this detailed information was meant for engineering analysis only, pilot attempts to decode the displays lead to confusion.

4.1.5 Actuator Interface

Considerable preliminary design work was spent on refining the interface of the triplex flight control system to the surface actuators, which were previously driven by the F-16 quadruplex analog flight control system. A system integration memo investigated seven possible reconfigurations using the triplex system, in terms of the reliability of each. In addition to having a reliable actuator interface, the DFCS was required to detect failures in the commands to the actuators before the actuators reconfigured for the failure condition. Additional test data on the actuators, which determined fault levels, allowed for an initial DFCS fault detection design. However, the fault levels dictated by the actuator characteristics were small enough that asynchronous sampling errors would cause nuisance failures. After several design iterations, the actuator interface requirements were finally met, illustrating that fault detection designs can require considerable effort and are dependent on device characteristics which may normally not be obvious.

The remainder of section 4.1.5 describes the ISA and the DFCS interface to the ISA; further detail can be found in Price and others (1984). Each of the seven integrated servoactuators (ISAs) accepts electrical commands from the flight control computers in three electrohydraulic servovalves (fig. 16). It converts these commands into a power-ram position, which then positions the respective control surface. Several significant functional characteristics are embodied in the design of each ISA:

1. A unique mechanical position and rate feedback scheme combines the feedbacks into a single input to each servovalve.

2. Three servovalves (SV1,2,3) are provided for redundancy; SV1 and SV2 normally share control of actuator position, while SV3 is held in standby.

3. Servovalve failure is detected by comparing servovalve first-stage pressures.

4. Self-contained hydromechanical failure detection and correction logic is incorporated for first failures of the servovalves or for the hydraulic system. A first failure of SV1 or SV2 will transfer control to the standby SV3. A first failure of SV3 will lock the servoactuator on SV1 and SV2 control.

5. Hydraulic system failure correction is given precedence over all servovalve failures. Servovalves SV1 and SV2 operate on one hydraulic system, and SV3 operates on the other hydraulic system.

6. Fail-safe capability is incorporated to allow the ISA to center mechanically upon receipt of an electrical command to the fail-safe solenoids from an external electronic model or monitor unit.

Each servovalve can be driven by either its primary or secondary coil. The primary coil of each servovalve is driven by a corresponding FLCC channel. In the event of an FLCC failure, the secondary coil is driven through the backup amp by one of the remaining good computers. As discussed earlier, fault detection of the ISA required a complex set of fault detection logic in the DFCS computers to monitor computer failures, failures in electrical commands to the servovalves, and the pressure of the hydraulic systems. The monitors allowed the actuator interface to operate after two failures.

In the case of a dual failure, a model of the actuator was run to detect any failures. If a failure was detected, the DFCS would send a signal to center the actuator, preventing a hardover command and subsequent loss of aircraft control.

The reliability of the DFCS and of its interface to the actuators was significantly better than the hydraulic and actuation system itself. Design of the interface to the actuators required information about the actuators which, at the time, was not available.

4.1.6 Electrical System Interface

Electrical power is required for aircraft control, since the aircraft has a full-authority DFCS. Reliability requirements of the DFCS were also applied to the electrical system. Electrical power from five sources provides the redundancy needed to ensure DFCS operation. The preliminary design for the electrical system is shown in figure 17. The DFCS only required 28 V dc power for operation. The five sources of power were provided by the 40 kVA primary generator and the 5 kVA emergency generator through ac to dc converters, a 500 VA permanent magnet generator (PMG) on the emergency generator, and two batteries.

In normal operation the primary generator provides 28 V dc power to the flight computers and maintains a charge on the two batteries. Loss of the engine or the primary generator will switch on the emergency generator. The PMG provides a limited 28 V dc through the 500 VA transformer rectifier (TR) unit for certain failures of the emergency generator. The power from the PMG TR and the batteries was supplied through three current switches to the

DFCS. The purpose of the current switch is to limit the output current so that short circuits in one channel would not affect others.

The major difference between the preliminary design and the design for flight test (fig. 18) was the inclusion of voltage detectors on any FLCC power line being fed by the emergency generator, including the PMG. During this time, it was found that emergency power failures on F-16 aircraft resulted in a failure mode which caused an overvoltage condition. The overvoltage condition inhibited proper flight control system operation.

When the AFTI design detected an overvoltage condition, the aircraft began operating on batteries. Battery size was increased to provide about 30 minutes of flight time.

A unique operating condition during flight test caused the PMG output to overvoltage. The cause and implications of this anomaly are discussed in section 7.2.1.

4.1.7 Selector-Monitor and Failure Manager

The names selector-monitor (S/M) and failure manager (FM) are derived from the two software components in which they are implemented. The S/M provides for: (1) signal selection, (2) fault detection, and (3) reconfiguration for discretes, sensors, controllers, output surface commands, and the actuator interface. The S/M software component works closely with the FM software for recording and analyzing failures.

The FM records and analyzes information provided by the S/M and provides for pilot resetting of failures. The designs of the S/M and FM were dependent on system architecture, asynchronous operation, and the unique characteristics of the hardware (sensors, controllers, actuators) being monitored. I have called this hardware-software design the fault-tolerant design. Redundancy management and fault detection, identification, and reconfiguration (FDIR) is also a common name. A part of the fault-tolerant design — built-in test — is discussed separately.

No formal tools were used to develop or analyze the fault-tolerant design, although analytical studies were performed to evaluate different algorithms. The S/M provided selection and fault detection for (1) input discretes (table 11), (2) analog inputs (table 12), (3) digital commands for each control surface, and (4) the actuator and its electronic interface. The input discretes fall into two categories based on the amount of switch bounce. The two categories are labeled by the settling time required by the discrete input.

The basic S/M approach was to use cross-channel information for signal selection and monitoring. After a channel had sampled its values, the information was sent in digital form to the other two channels for comparison (fig. 19).

The majority of analog signal values was obtained using a good-channel-average (GCA) algorithm with selection of discrete values by a majority vote. The GCA algorithm is summarized in figure 20. Any value which differs from the other two by a preset threshold is declared failed. The fault-detection algorithms allowed for setting unique failure thresholds and persistence times for each of the inputs. In certain failure cases, such as dual failures, a model was run to provide the needed information to resolve failures. Actuator and leading edge flap (LEF) models were both used to resolve dual failures.

The most complex aspect of the S/M is the actuator interface and LEF fault detection algorithms. Multiple monitors were needed to provide proper selection and fault detection of the unique hardware. Included are wrap-around, output electronics, lock, and centering monitors.

The failure manager component provides much of the intelligence behind the dual-fail-operate design aspects. Using a hierarchical structure (table 13), the failure manager would analyze individual failures and fail higher or lower level devices accordingly. For example, a failure of the analog-to-digital converter would result in logically failing all devices below it (see 2.1 through 2.4, table 13). In another example, if a pitch rate sensor failed first in channel A and an inverter failed second in channel B, the FM would attribute a second detected failure of a pitch rate sensor to the failed inverter — and not to a disagreement between the two sensors. Therefore, all three computers would use the last pitch rate sensor from channel C. If the second failure of the same type sensor cannot be resolved, the control laws will be reconfigured so that the sensor information is not needed.

The S/M used on the digital commands for each control surface (the result of the control law computations) deserves special attention. This software detected partial failures of computers that resulted in wrong computations of a given surface command. Individual computation failures were allowed for each of the seven surface commands. Failure of three or more surface computations resulted in the FM declaring an entire channel failed, if not already detected by hardware monitors.

Failure thresholds for the surface commands were originally set at 15 percent of the full-scale command. Persistence time, the time required for the error to persist before declaring a failure, was four iterations or about 62 msec. Owing to command errors introduced during dynamic maneuvers, the failure thresholds had to be increased. A variable threshold was used, with the rate of change of the surface command added to the 15-percent baseline. If the surface moved 5 percent of full-scale in the previous iteration, the failure threshold would be 20 percent. Failure thresholds were independently calculated in each channel. Channel signal selection for output surface commands was a function of channel and hydraulic failures. In the nonfailed state all three channels used channel B's value. This was required to minimize the errors between asynchronous channel commands which would otherwise be detected by the actuators.

The fault-tolerant design for detecting failures of individual computers is summarized in figure 21. The three methods for failing the entire computer processor and input-output controller include watchdog timer, consensus of other two channels, or self-test failure when it is run to resolve second failures. In the case where three or more surface calculations fail, the computational portion is failed but the input-output controller runs independently, supplying sensor information to the remaining two channels.

Tools to study the effects of different fault-tolerant designs are lacking. Different selection and fault-detection algorithms were designed and analyzed using analytical studies. Simulation or emulation of the fault-tolerant design was not available to determine the effects on reliability or interaction with control law algorithms.

The hierarchical structure used to improve fault tolerance by resolving second failures is a novel design. Failure modes and effects testing demonstrated that this approach is a valid method to increase fault tolerance. The hierarchical structure provides designed-in knowledge of the system. This knowledge provides the information needed to resolve high- and low-level failures.

Memory parity checking was included in the computer hardware. A parity error interrupt occurred when bad parity was detected. However, the fault-tolerant design did not consider parity errors. The interrupt and memory address were saved and processing continued. This approach of ignoring hardware failure indications, unless they resulted in output command failures, raised concerns about latent failures. The effect on system reliability when this type of latent fault is allowed was not modeled.

The fault-tolerant design was impacted heavily by the asynchronous computer operation. Errors between channels in the input sensors and controller, owing to time-skewed sampling and dynamic conditions, caused two main problems:

1. Errors between channels forced the failure thresholds higher. The 15 percent of full-scale plus rate value allowed large failure transients. Horizontal tail transients corresponding to the 15-percent failure transient are 3.75°. At low altitude, high-speed conditions, the aircraft's normal acceleration transient would exceed 3 g, well beyond that called out in the specifications (table 27).

2. Errors between the channel inputs were passed through the control law calculations to the outputs. In order for the actuators to accept the commands without having the hydraulic system voting, an output command selection method was needed. The method required all three channels to choose one value — channel B's value is used in the non-failed case — to drive the actuators. The triple system appeared as a single system with one channel controlling the aircraft within the failure thresholds.

4.1.8 Built-In Test and Memory Mode

The built-in test (BIT) is run prior to each flight to ensure the integrity of the DFCS. The BIT is also used in maintenance procedures to isolate faults to the line replaceable unit (LRU) level. Memory mode was another pilot option available only on the ground. It allowed the pilot to give the flight control system a memory address and obtain a readout of the three computers' corresponding values.

The BIT consisted of four major test categories: (1) input, (2) computation, (3) output, and (4) failure logic. The three channels had to be synchronized to get valid BIT results.

Input testing ensured proper operation of sensor and input conversion hardware. Null failures, which would remain latent until aircraft motion allowed for fault detection, were detected by BIT. Hardware input signal conversion failures were separated from sensor failures by injecting input signal biases into the hardware under BIT software control (fig. 22). Passing this test, BIT would then torque the sensors to test for faults. Tests for the avionics multiplex bus (AMUX) and cross-channel data link, both serial digital buses, were also performed.

Computational tests include those for the CPU, RAM, and read-only memory (ROM). Output tests are run by BIT for the seven actuators, LEF, and output conversion hardware (digital to analog). Testing also included detecting null or passive failures of components that are used only in the event of failures. Testing the ability to center an actuator is an example, even though it was required only when multiple failures occurred.

The BIT requirements for testing the failure logic is similar to the actuator centering example. Although the failure logic is only used when single or multiple channel failures occur, latent failures in this logic circuitry could prove catastrophic. The BIT checked for passive failures not detectable by the inflight fault-detection routines.

The BIT operation suspends all control law and fault-detection routines required for flight. Therefore, two lockout methods were used to ensure BIT and memory mode would not operate in flight. The weight on wheels switch and lack of main wheel spin-up (less than 28 knots) were required to allow activation of the modes. Both lockouts had triple signal redundancy.

The BIT and memory mode were activated through the multipurpose displays (fig. 14). The DFCS test page display provided BIT and memory mode initiation and display. The BIT required two and a half minutes to complete. The memory option allowed for displaying memory values of all three computers and was used for troubleshooting BIT failures. Communication to the DFCS to initiate BIT was over the AMUX bus to one DFCS channel. The cross-channel data link was used to inform the other two channels. The time of BIT initiation varied in three channels owing to asynchronous operation and was a function of computer skew.

The BIT design is a comprehensive, structured set of tests that ensure hardware integrity prior to takeoff. The mechanization document to the software group contains over 180 pages in the BIT section. Memory mode proved to be a valuable troubleshooting tool.

4.2 Control Laws

This section includes an overview of the methods used to develop the control laws and the features designed into each control mode. The control laws are described to provide the reader with an overview of the system.

The control law designs provide for mission specific flight modes and standard and decoupled aircraft control (fig. 23). Mission specific modes are standard normal (SNRM is for takeoff, refueling, and landing), standard air-to-air guns (SAAG), standard air-to-surface guns (SASG), and standard air-to-surface bombs (SASB). A decoupled version of each standard mode is available through the decoupled mode selection switch on the right-hand controller. Pilot control is through the side stick, force stick, and the rudder pedals, and decoupled pitch axis control is through a modified throttle controller that twists.

4.2.1 Control Law Development Process

The control laws were developed using four different, and progressively more detailed, methods. The initial design was obtained using linear continuous models. The longitudinal axis used linear quadratic synthesis for its initial design. The second design method included discrete sampling effects of the digital system with the linear models. Criteria for natural frequency, damping, and phase and gain margins were used to evaluate the designs. Nonlinearities were included in the third design method which used a nonreal-time batch simulation. The final design evaluations used man-in-the-loop real-time simulations. Evaluations were centered on handling-qualities criteria, tracking tasks, and weapon delivery accuracy. Anderson and Frank (1984) provide additional information on the control law developments.

4.2.2 Control Law Design

The eight mission specific standard and decoupled control modes are implemented in five longitudinal control structures and three lateral-directional structures. An overview of each control law, the mission specific modes applicable, and primary design features is given.

Features of all the longitudinal control laws are as follows:

1. Neutral speed stability,

2. Drag modulation,

3. Near constant stick force per g,

4. Departure prevention,

5. Angle of attack and lead factor limiting,

6. Optimal flap scheduling on angle of attack,

7. Maneuvering flaps, and

8. Structural filters.

The first longitudinal control law structure is used only in the SNRM. The SNRM provides aircraft control for takeoff, in-flight refueling, formation flying, and landing. If SNRM was not previously selected for landing, it is automatically selected when the landing gear handle is lowered. The SNRM control laws implement a pitch rate system gear down and a G-command (GCMD) system gear up.

The second longitudinal control structure is used when sensor failures force reconfiguration of the control structure to account for loss of a feedback. Termed the reconfiguration mode, it allows for operation with complete loss of pitch rate, normal acceleration, angle of attack or air data sensors. Multiple reconfigurations are not acceptable. Normal acceleration and angle-of-attack reconfigurations used the SNRM pitch rate system with the normal acceleration or angle-of-attack value set to zero.

Because the airplane is statically unstable, pitch rate reconfiguration required estimating pitch rate based on elevator command. The complete failure of air data used to schedule control system gains resulted in standby gains. The standby gains were a predetermined set of gain values that provide adequate stability and control throughout the envelope, and only change as a function of the landing gear position.

The third longitudinal control law was used in the SAAG and SASG modes. The major difference from the SNRM mode was that pitch rate is used as the main feedback, providing for better tracking, improved flightpath control, and reduced tracking errors in turbulence.

The fourth longitudinal control law was used in the SASB mode. This mode was similar to the SNRM. The SNRM gains are changed as a function of Mach number and altitude to give SASB mode (fig. 24). In order to improve flightpath response, the maneuvering flap gain is increased.

The final longitudinal control law was used by all the decoupled modes. Three types of decoupled longitudinal controls were available — pointing, translation, and direct lift. Pointing generated pitch rate without generating normal acceleration; translation generated normal acceleration without generating pitch rate; direct lift generated a combination of pitch rate and normal acceleration without changing angle of attack.

Flightpath maneuver enhancement (FPME) and pitch rate maneuver enhancement (PRME) were two coupled (conventional) control features commanded with the sidestick. These were the only features available on the right side-stick, and they operate independently or simultaneously with throttle-twist decoupled mode inputs. The FPME was tailored for the cruise and bombing tasks because the mode provided a responsive deadbeat normal-acceleration response. The PRME was tailored for air-to-air tracking and air-to-ground strafing because this mode required deadbeat pitch rate response. All longitudinal modes operated at a 64 Hz iteration rate.

The first lateral-directional (LAT-DIR) control law was used in the SNRM only. Features include

1. Roll rate command prefilter which was quicker for stopping roll rates than when initiating them,

2. Lateral acceleration, roll rate, and yaw rate feedbacks,

3. An aileron-rudder interconnect,

4. A roll-rate and angle-of-attack interconnect,

5. Gun-firing compensation, and

6. Structural filters.

The second LAT-DIR mode was the reconfiguration mode, which is based on the SNRM. Roll rate, lateral acceleration, yaw rate, and air data reconfigurations were included.

The last LAT-DIR mode was used by SAAG, SASG, SASB, and all the decoupled modes. Three decoupled control options were available in this control law structure — direct sideforce, pointing, and translation. In all three standard modes, direct sideforce was commanded via the rudder pedals. The relationship of decoupled modes to decoupled control options is shown in figure 23. The different control options were switched in the one control structure through the use of software. The switches, nearly 20 in this control law structure, create the required submode structures that represent each decoupled option.

The LAT-DIR control modes operated at two iteration rates. Control inputs, roll stick, and pedals were shaped and filtered at 32 Hz. Feedback paths and interconnects operated at a 64-Hz iteration rate.

The number of control modes and the decoupled capabilities required resulted in a complex set of control structures. In the longitudinal modes, where the linear quadratic synthesis (LQS) design methodology was used, gain schedules requiring double interpolations were needed (fig. 25). Gain N_1, the stick feed forward gain, requires interpolation of altitude and Mach number.

From a system engineering viewpoint, the complex control structures were a boiling pot of ingredients. These ingredients included outputs from the selector monitor and submode switches based on pilot selections and aircraft configuration. Besides these inputs to the control structures, the control structures themselves provided surface commands to the output fault detection routines. The control law design presented a formidable task in qualifying the system.

4.3 Digital Flight Control System Software

The process used to obtain the software design and an overview of the software design itself are presented in this section. A discussion of the test approach used for the software is given in section 5.

The process used in obtaining the design is summarized in terms of the supporting documents, the configuration control process, the software support tools used, and the implementation language for the flight computers. The method and parameters used to track the progress of software design are given, and a brief description of the structure and content of the software design is presented.

4.3.1 Software Development Process

The process used to develop the DFCS software was based on MIL-STD-483 (U.S. Department of Defense, 1985a) and MIL-STD-490 (U.S. Department of Defense, 1985b). Documentation included the computer program development specification (CPDS), computer program product specification (CPPS), and an assortment of test plans and procedures.

The complexity of the DFCS, coupled with the need to state clearly the functional design requirements, control laws, fault-tolerant design, and timing constraints, resulted in an additional document. The software mechanization document, not required by military standards, provided the interface between the system designers and the software development team. In addition to describing the seven top-level software structures (fig. 26), the document included a section on the software needed to test the DFCS and a description of the hardware.

A few examples will help one understand the role this document played in software development. Table 14 shows the detail provided in the mechanization document for describing the selection and monitoring of analog input signals when all three inputs are valid. Information is provided for all nine failure conditions, how to increment or decrement persistence counters, and which signal selection to use. Figure 27 shows the software mechanization for a typical control law module from over 50 such modules. Note that filters are given in Z-transform representation. This level of detailed information was required before software design could begin.

An overview of the software design process is shown in figure 28. It shows the software development activity from design to release of the software to the test team. Depending on the type of error or redesign required, the mechanization document (not shown) would also be updated.

The FLCC were programmed in assembly language. The instruction set was simple, having just over 80 instructions. The program was controlled using jump and skip instructions. Only three jump instructions were available, with all conditional control (for example, less than, equal to zero) being achieved by the use of the 24 skip instructions. The instruction set was unique and did not represent any standard microprocessor instruction sets. All calculations were done in single or double precision fixed-point formats. The programmer can enable a unique hardware function called saturation arithmetic. This did not allow overflows to occur but saturated the value at its maximum scaled limit.

The first milestone in developing the software was the critical design review. The purpose of this design review was to show how the functional system design for the control laws and fault-tolerant design were to be implemented. However, an iterative cycle developed as the functional system design was implemented into the software. Real-time and memory constraints of the software implementation forced changes in the functional system design. Changes of the fault-tolerant design in the areas of the output selector monitor and the ISA fault detection were required. All totaled, a 4-month schedule slip was incurred before the critical design review could be held.

A schedule showing the critical path for the software development first appeared at the critical design review. The percentage completed and the scheduled date for completion were given for (1) software mechanization, (2) software design, (3) software code, (4) integration with hardware, (5) unit and module test, (6) stand-alone verification, and (7) integrated system validation. Software real-time and memory use were also tracked.

4.3.2 Software Design

A top-down structured approach was used for the software. The highest software structure was the computer program component, followed by the module and the unit (fig. 29). The data base was also a structured design (fig. 30). Most of the software components — system monitor, selector monitor, failure manager, start-up and restart, and AMUX processor — provide the software portion of the fault-tolerant design (see fig. 26). The remaining two software components are the control laws and the executive.

The detailed software description document required by the U.S. Air Force is the computer program product specification (CPPS), containing the software design. It was used to translate the system design into a format appropriate for the programming team. The CPPS, although essential for the software designers, was of little value to system designers and users. Even with the addition of the software mechanization document, following a design requirement through the mechanization and CPPS documents was difficult for technical managers and system users, and only a few talented engineers could track it. Another problem with the CPPS was its size and the arduous manual method required to update it. The first complete CPPS was released at the critical design review and required eight 3-in. binders. The next complete release of the CPPS was not available until 2 years later.

The software mechanization document was required. It helped the designers and users to understand the design. The mechanization document used a combination of English, if-then-else psuedo language, tables, and graphs. Like the CPPS, the requirements in the mechanization document used no formal method to determine their completeness or correctness; that was the job of the design reviews.

5 SYSTEM-SOFTWARE QUALIFICATION AND DESIGN ITERATIONS

The qualification of the design presented in section 4 is discussed in this section. System-software qualification refers to the testing of the design following implementation. Errors or discrepancies found during test are corrected with a new design and the cycle begins again.

Qualification of the DFCS as a system and the testing and redesign of the software are emphasized. Qualification of the hardware in terms of environmental testing is not covered. The methods and problems encountered when integrating functional requirements (such as control laws and fault tolerance) with the system architecture (such as asynchronous computers) are highlighted. The three areas discussed in the preliminary design section — system architecture, control laws, and fault tolerance — all go through the qualification and design iteration process. Since the major points deal with the process as a whole rather than how it applies to each of the three functional areas, this section is structured in terms of the qualification activities.

An overview of the iterative process, design documentation updates, testing performed, and environment for the testing is shown in figure 31. The first verification test is performed by the programming team and ensures that the software is implemented per its design specification. Configuration control is maintained by the software team at this design iteration level. The second verification test is done by a group independent of the software team, using the mechanization document. This is the first test with an entire software package operating in a test environment with the triplex flight control hardware. Configuration control was handled by a board that included disciplinary engineers — control law designers, fault-tolerant designers, and hardware and software designers. This configuration control process was also used to resolve system validation discrepancies. The last iteration cycle results from validation testing, which ensures that the system design is correct, not if the software follows the specification. The system software is validated with all the flight control hardware operating in real time, with an aerodynamic simulation. Special provisions to induce failures into the system, such as sensor failures, are included for failure modes and effects testing (FMET). A mockup of the cockpit with flight controllers and displays is used to support flying-quality evaluations. Some validation testing requires the actual airframe. Structure coupling and electromagnetic interference and compatibility (EMIC) were tested with the aircraft. Additional details on the system-software qualification can be found in Gordoa and others (1984).

5.1 Schedule

Before discussing the qualification process, a review of the schedule and the parameters tracked for the qualification task will be helpful.

The schedule for qualifying the AFTI system is shown in figure 32. The release of the software package, after unit verification by the software development team, was scheduled for November 1980, only 2 months after the critical design review. Verification by an independent test group was scheduled for February 1981. System validation was scheduled for completion by April 1981, with the first flight in July 1981. The qualification process required an additional year to complete beyond the original schedule.

A history, taken from the program's status reports for each of the three qualification phases, is shown in figures 33, 34, 35.

The percentage of completion and the predicted amount of schedule slip were obtained from contractor status reports. Software coding and unit test were the only tests to show 100-percent completion. Software verification reached 90-percent and system validation 30-percent completion in the last status report which tracked them. All testing did reach 100 percent at the actual completion date shown on the figures. The original and actual completion date is shown for each test. Once the previous testing was completed and its discrepancies corrected, the next testing phase began to succeed.

The primary issue regarding the schedule was its optimism. The main flaw in estimating the schedule was a belief that once released from the software group, verification and validation testing of the software package would be completed nearly error free. The iterative nature of testing and redesign was not acknowledged. The iterative nature is shown by the number of software versions released, 14, and the number of mechanization change notices, over 600, required to achieve a software release acceptable for first flight.

Estimating schedules for software-driven systems of a research nature is difficult. A few of the parameters that affect schedules include

1. System complexity,

2. Programming language and methods,

3. Development tools,

4. Software program size,

5. Staff experience,

6. Testing requirements, and

7. Required documentation.

All the foregoing items will affect the number of errors in the design and therefore the number of design iterations. For flight or life critical applications, testing must be thorough. Thorough testing of a design is time consuming, testing multiple designs even more so. Documentation is a large effort and cannot be overlooked when determining schedules.

5.2 Software Verification

The methods and tools used for software verification after its release by the software development team are discussed in this section. Verification is defined as the testing performed to ensure the software is implemented with respect to the mechanization document. The term stand-alone testing was used by the contractor because of the configuration of the DFCS. The DFCS stands alone from the avionics systems, cockpit interface, and the simulation, for this testing.

A general description of the verification test environment includes

1. Complete software package (including unique test software),

2. Triplex flight computers, and

3. Test support equipment.

5.2.1 Verification Test Plan

A preliminary test plan was released in March 1980 for government review. After several review cycles and coordination meetings, the final plan was released in August 1980. The review cycle provided an excellent interchange where the majority of government and contractor concerns were resolved.

Contents of the verification test plan section are shown in table 15. This test plan represented a thorough description of the purpose, the system under test, qualification requirements, success criteria, test implementation, and configuration control.

The purpose of verification testing was to

1. Verify the software independent of the software development group,

2. Detect errors in the software interface to the computer hardware,

3. Detect errors with the software operating in a triplex hardware configuration, and

4. Provide a solid software package to support system validation testing.

Although not an objective of verification testing, system design errors were found and corrected.

A separate verification document contained detailed testing procedures used by the testers. This document was formally published in October 1981. During this time, several preliminary software releases were made. Preliminary verification testing began in March 1981 and went through October. This testing detected errors while allowing for the development of the test procedures used during formal verification. Nearly 150 discrepancies were found and corrected during this preliminary testing.

5.2.2 Verification Support Equipment

The support equipment used for verification testing is shown in figure 36. The primary equipment is the engineering test support equipment (ETSE) test complex furnished with the flight computers. To provide improved test documentation and some automation of the redundancy management testing process, test facilities were upgraded to allow for an ETSE real-time memory monitoring of the flight computers, and a redundancy management test signal generator was implemented. An on-line capability for recording MIL-STD-1553 multiplex bus data was also available.

5.2.3 Verification Tests

The 13 verification tests are listed in section 4 of the Test Plan, table 15. The unique characteristics of each test are discussed briefly.

5.2.3.1 Avionics multiplex bus data interface test. The avionics multiplex bus (AMUX) data interface test verifies that the software performs the following functions:

1. Provides two-way communications between DFCS and other systems for data input and output;

2. Provides cross-linking of data from the receiving FLCC to the other two FLCC;

3. Allows data transfer over either AMUX A bus or AMUX B bus while retaining data consistency;

4. Maintains information on current data and location of the new data.

5.2.3.2 Gain scheduler test. The gain scheduler test verifies that all scheduled gains (for all flight control modes) that are a function of slow-moving air data are computed 4 times per sec.

Some gains are also a function of aircraft configuration, that is, gear down. Testing for these conditions was also included. Unlikely, but physically possible conditions, such as negative static pressure, were also tested, uncovering some unique errors.

5.2.3.3 Control law frequency test. The control law frequency response test verifies that the open-loop response provides the required system gain and phase margins.

This test was actually a validation test and did not completely verify the correctness of the code. Testing of limiters, signal shapers, and logic conditions was not covered by this approach. Control law verification became a test issue as a black box approach was used rather than a detailed test of each control law function. This is discussed in section 8.

5.2.3.4 Control mode selection and transition response test. This test verifies the open-loop control mode and submode switching and transition response characteristics of the DFCS.

5.2.3.5 Analog input selector and monitor test. This redundancy management test verifies the capability of the DFCS to monitor and select the control law input data in the presence of single and multiple input failures. Verification included that for fault analysis, fault recording and reporting, and reconfiguration and reset functions of failure management. A large portion of the S/M testing was automated.

5.2.3.6 Discrete input selector monitor operation. This redundancy management test verifies the capability of the DFCS to monitor and select discrete input data for control law computations in the presence of single and dual input failures. Verification includes handling of discrete input inconsistencies and the fault analysis, fault recording and reporting, and reconfiguration and reset functions of failure management.

Simultaneous nose up and nose down trim conditions are types of inconsistencies resolved by the software.

5.2.3.7 Integrated servoactuator test. This redundancy management test verifies that the integrated servoactuator (ISA) subframe and frame cyclic monitors of the DFCS provide valid output commands to each of the primary airframe control surfaces in the presence of computational and electronic drive failures. Verification testing includes the corrective action, fault recording and reporting functions of failure management. Tests also include verification of ISA centering and failure reset under appropriate conditions.

5.2.3.8 Leading edge flap test. This redundancy management test verifies that the leading edge flap (LEF) cyclic monitor of the DFCS provides valid output commands to the dual LEF drive system in the presence of computational and electronic failures. Verification testing includes the corrective action, fault recording and reporting functions of failure management, and the capability to lock the LEF drive(s) and failure reset under appropriate conditions.

5.2.3.9 Long power outage test. This test verifies that the DFCS restores the FLCC to normal operation after a long power outage (>50 msec).

5.2.3.10 Short power outage test. This test verifies that the DFCS OFP restores the FLCC to normal operation after a short power outage (<50 msec).

5.2.3.11 Memory and duty cycle reserve test. This test verifies that the software provides 30-percent memory reserve and 25-percent duty cycle reserve. In addition, timing data were obtained with regard to completion of cyclic tasks and ISA subframe monitoring to determine the adequacy of frame reserve time and the software executive scheduling of tasks.

5.2.3.12 Built-in test. The built-in test (BIT) verifies that the DFCS provides the capability for verification of system integrity (no hardware failures) prior to flight, and fault isolation to the line replaceable unit level during maintenance operations. The original approach to insert hardware failures to test the BIT software was abandoned owing to scheduling constraints. BIT was tested by patching the software to indicate failed conditions.

5.2.3.13 Flyable hardware retest. Flyable hardware retesting was a major issue during planning but became a moot point. Originally, the software, in following its schedule, was well ahead of the hardware development. It was believed that only breadboarded computers would be available for testing. However, the DFCS software release slip allowed time for flyable hardware delivery. Testing on the flyable hardware was the best approach and was probably the only good result of the software slip.

5.2.4 Reverifying the Design Iterations

The tough question for verification is how much retesting is required to verify that the change does only what it is supposed to do? Testing what a change is supposed to do is fairly straightforward, but how much additional testing of other software components is needed?

As mentioned earlier in section 5.2.1, preliminary verification began in March 1981 and formal verification began in November 1981. From the start to end of formal verification, June 1982, 14 software releases were made, a rate of nearly 2 per mo. Although the last releases had considerable testing, no single release had every test run on it.

The approach for retesting the final release prior to flight test, and also those releases during flight test, was to test each change made and then run a software acceptance test procedure (ATP). Reverification procedures contained a variety of tests for each of the software components (table 16). Two other considerations are worth noting: (1) structured and controlled coding techniques help to reduce errors, and (2) the follow-on system validation testing helped to detect errors caused by redesign.

5.3 System Validation

The system validation test demonstrates that the system, as a whole, performs as expected in the user's environment. For the most part the desired system performance is usually not specified to the detail needed. In the stability and control discipline, the performance is specified with more detail than in other disciplines. For example, flying-quality criteria exist, such as the allowable time constant of the spiral mode. However, detailed criteria for other areas, such as the fault-tolerant design, is almost nonexistent. All aspects of the system validation process, including planning, test environment, and validation test and retesting, are discussed in this section.

5.3.1 Validation Test Plan

The types of validation testing were spelled out in seven separate documents. Some of the test plans were written by the DFCS test group, others by the simulation group. The test plans were

1. Integrated system testing of the DFCS software (DFCS test group),

2. Controls and displays test plan (simulation),

3. Flying-qualities test plan (simulation),

4. Failure modes and effects test plan (simulation),

5. Flight control system ground test plan (on aircraft tests),

6. Electrical system test plan (on aircraft tests), and

7. Electromagnetic compatibility test (on aircraft test).

The last three items are validation tests performed on the aircraft and are listed for completeness. They will not all be addressed in detail. Flight test is the final validation test and is covered in section 7.

The test planning effort was useful. In many cases the testing and test documents required by the U.S. Air Force were redundant, and owing to a general misunderstanding of what validation testing is, many duplicate tests were created. This duplication occurred most often in areas least understood, such as the fault-tolerant design. The test planning effort helped to reduce the overlap in testing. It provided an excellent interchange between the government and contractor.

5.3.2 Support Equipment

The validation test support equipment includes all the equipment used for verification testing. Additional test equipment included the aircraft simulation, visual system, and the avionics interface (fig. 37).

The aircraft simulator was comprised of digital models for the aircraft's aerodynamic characteristics, an engine model, and models of the actuators. The actuators were also modeled using analog circuitry; these analog models were usually used during testing.

The primary visual display was a 4- by 4-ft projected display. The display and mock-up cockpit were housed in an 18-ft dome structure. Visual displays for air to air, air to ground, and take off and landing task were available. The air-to-air target could be set up with several preplanned maneuvers. The avionics equipment included an FCC, SMS, and PDG used to provide multipurpose display information in the cockpit.

Since validation testing is designed to show proper system operation, the environment in which the system is used must be modeled in detail. The aerodynamic model used is based on the F-16 wind-tunnel data which was improved with canard and dorsal fairing effects from wind-tunnel tests and also with F-16 flight-test data.

Maximizing use of the real avionics subsystems helped to provide the actual environment. Using the avionics hardware also provided an opportunity to resolve the subsystem interfacing problems that occur during validation testing.

5.3.3 Validation tests

A detailed look at the validation tests performed on the DFCS is undertaken in this section. Areas of overlap and secondary benefits from the testing are discussed.

5.3.3.1 Integrated system testing of the DFCS software. Integrated system testing, which included testing of more than just the software, was handled by the flight control system test team. The tests are shown in table 17. A brief description of the goal for each test follows.

The BIT validated that the DFCS provided the capability for verification of system integrity (no hardware failures) prior to flight and fault isolation to the LRU level during maintenance operations. The verification test consisted of loading software drivers that made BIT think hardware failures occurred. The validation test for BIT consisted of running it prior to each day's operations. This only validated BIT to the extent that BIT did not falsely detect errors and correctly detected the few failures which did occur. BIT is discussed further in section 7.

The mode selection and display test validated the DFCS, FCC, SMS, AMUX and MPD interfaces required for pilot mode selection and display. Additionally, the test validated the pilot's capability to communicate with the DFCS through the MPD and applicable cockpit switches.

This testing addressed controls and displays in detail — one area where testing overlapped — in the simulation test plan.

The control law frequency response test validated that the DFCS provided satisfactory closed-loop frequency response characteristics. The test provided validation of closed-loop phase and gain margins. A frequency generator and X-Y plotter were used. The phase and gain margins were read from the X-Y plots generated.

The step response test validated that the DFCS control laws and executive control of tasks provided satisfactory multimode transient response characteristics, owing to pilot controller inputs at 1-g and trimmed elevated-g conditions over the entire flight envelope. The tests included switching transient tests from standard to decoupled modes and vice versa. Test inputs for the pilot controllers were computer generated with the aircraft response compared to results obtained by the batch simulation. The batch simulation used independently modeled control laws.

The flight scenario test validated that the DFCS provided satisfactory control law performance during takeoff, climb, acceleration, and landing conditions. The test profiles were performed with and without stores, drag modulation, and optimum flap schedule. This type of testing was also covered in the flying-qualities simulation tests.

The analog input, single failure tolerance test validated the ability of the DFCS to maintain full operational capability during transient and latched single failures of analog inputs. This test also validated the ability of the pilot to reset failures, and that the resultant failure and failure reset transients were within acceptable levels.

The analog dual-like failure tolerance test validated that the DFCS provided safe recovery and landing capability from dual-like analog input sensor failures during high-performance maneuvers. The test also validated the capability of the pilot to reset transient failures that did not persist beyond seven iterations and that he was unable to reset permanent failures. A landing in the applicable reconfigured mode was included.

The ISA monitor test validated that the DFCS software provided the ability to maintain full operational capability during transient and latched single failures of the ISA interface system. The test also validated the capability to reset failures and center an ISA's surface under appropriate conditions and land the aircraft safely.

The LEF monitor test validated that the DFCS software provided the ability to maintain full operational capability during transient and latched single failures of the LEF interface system. The test also validated the capability to reset failures and lock an LEF drive under appropriate conditions and land the aircraft safely.

The analog multiple dissimilar input failure test validated that the DFCS software performed properly in the presence of multiple dissimilar analog input failures during high-performance maneuvers, including landing.

Note, the preceding five tests were also covered in the FMET.

The stress test demonstrated that the DFCS software executed correctly under extreme flight conditions and maneuvers in the presence of compound failures and improbable or erroneous inputs. The test was comprised of four major phases. In phase 1, coupled, large amplitude roll-yaw maneuvers were performed at flight conditions as close to the 1-g flight envelope boundaries as possible (fig. 38). Phase 2 consisted of performing coupled pitch-roll-yaw maneuvers while flying a high-speed loop which encompassed the entire Mach-altitude range of the vehicle (fig. 39). In phase 3, several landings and take-offs were made under extreme conditions. Phase 4 consisted of a sequence of HUD mission phase switch closures executed in a rapid manner at one flight condition while the vehicle was being externally forced in all three axes by a sinusoid stick input.

The power outage-restart test validated that the software restored the FLCCs to normal operation after a short power outage and a long power outage. This testing is also addressed in FMET. The objectives of the test also included the following:

1. FLCC without power is voted off-line.

2. IBU is automatically engaged when all three FLCCs incur a simultaneous power outage.

3. Pilot can reset a failed FLCC after power is resumed.

4. In case of power failures in two FLCCs, the remaining FLCC controls the aircraft.

5. Correct fault recording in nonvolitive memory and annunciation on MPD is achieved.

6. Manual engagement of IBU is permitted.

The control law gain margin test validated that the DFCS provided the required gain margin with the flap, elevator, and rudder loops individually open at the actuator and the other surfaces closed loop. This test was run for each control mode at a flight condition of Mach 0.9 and sea level. Pass-fail criteria were based on those obtained from linear analysis. Gain increases were made by increasing surface effectiveness in the simulation.

5.3.3.2 Controls and displays validation. The controls and displays test is a simulation test developed in part by the human factors engineers. Four tests were performed with project pilots to evaluate the cockpit controllers and displays.

The interior lighting evaluation test was the first test conducted as part of the controls and displays test. The objective of this test was to determine how well the MPDs, HUD, and mission phase lights meet operational requirements under various levels of ambient illumination.

A qualitative assessment of the utility of the MPDs, HUD, and mission phase lights was made by the pilots for high and low levels of ambient illumination.

The controls test had the pilots perform a single-axis tracking task of an air-to-air target with a fixed pipper using a single controller, either the twist throttle, rudder pedals, or side stick controller. They were required to fly the fixed reticle on the head-up display to the target represented by the target designator box, also on the head-up display. The duration of each trial was 30 seconds, and five trails were run for each of the flight control modes. This test concentrated on the human factors aspects of the cockpit controllers rather than the aircraft's flying qualities data discussed earlier.

The head-up display (HUD) is the pilot's primary flight instrument. The

AFTI's HUD is an improved version over the F-16 fleet. The field of view (display size) was larger and included system status information in addition to aircraft state displays (altitude, heading, and airspeed). The HUD symbology displayed changes as a function of aircraft mission and subsystem modes.

The HUD symbology was evaluated by each of the program pilots by flying seven different flight-mission phases in both coupled and decoupled modes. The different flight-mission phases and flight control modes provided the conditions for using or viewing the various HUD symbologies.

The objective of the integrated cockpit test was to validate the cockpit design in a dynamic situation. This task was structured to analyze the pilot's ability to maintain specific flight parameters and complete various mission related tasks. The following aspects of the pilot-vehicle interface were examined:

1. Logic, legibility and operation of the MPDs,

2. Pilot manual and visual access to cockpit controls and displays,

3. Side stick as a controller for coupled and decoupled flight,

4. Linear throttle as a thrust controller,

5. Throttle movement for cross coupling with the decoupled flight control system,

6. Switches on the stick and throttle for cross coupling with the flight modes,

7. Trim controls on the flight control panel,

8. Mission phase selection and status,

9. Warnings,

10. Threat warning enunciation and display,

11. Pitch limit operation, control, and display,

12. CCV engagement, disengagement, and display,

13. Manual pitch override operation procedure, and

14. IBU status, manual and automatic engagement, and manual disengagement.

5.3.3.3 Flying qualities. Before discussing the flying-qualities validation test, it is worth reviewing the history of this testing as used in the AFTI development. Note, that this validation test was used not only at the end of the development, but throughout the development, allowing early detection of design errors as shown by poor flying qualities. The flying qualities validation test is documented in more detail in Anderson and Frank (1984).

Three flying-qualities demonstrations were held before the final validation test. These demonstrations were requested by the flight test team before each major program review. The control laws evaluated were modeled in FORTRAN and run with the digitally modeled aircraft dynamics. Visual displays and a mock-up cockpit were part of the simulation. An example of the type of tasks and the modes evaluated is shown in table 18.

The first two prevalidation tests allowed for the early detection of design errors. Since the test environment did not include DFCS hardware and software, the tests cannot be classified as true system validation tests. However, the following benefits must be recognized:

1. Early detection of design errors provides an improved product, perhaps at a lower cost;

2. Provides critical user and flight test pilot feedback to designers; and

3. Provides visibility into control law design for all program participants.

Also, because of the test environment and the inherent limitations of flight simulations, some pitfalls must be recognized and avoided, including the following:

1. Modeling errors can give indications of design errors which do not actually exist. Control law modifications to correct these phantom errors can adversely affect flight test. Examples include time delays, both real system displays and those perceived by the pilot because of display cues, and the lack of nonlinearities, such as those in the actuation system.

2. As a pilot evaluation without the safety implications and stress situations of actual flight, certain tasks, such as landing, may not provide credible results.

The third flying-qualities validation test was held before the flight readiness review. This flying-qualities test used flight hardware and completely verified software. The flying qualities test is summarized as follows:

1. Evaluates flying qualities in SNRM mode,

2. Demonstrates satisfactory flying qualities in the task tailored modes (SAAG, SASG, SASB),

3. Demonstrates satisfactory flying qualities in the decoupled modes (DNRM, DAAG, DASG, DASB),

4. Determines that no deficiencies in flying qualities or handling qualities limit flight safety or the capability to perform the intended missions,

5. Develops flight test prediction data,

6. Examines DFCS modes and flight conditions that will not be flight tested but could be entered by improper pilot mode selection,

7. Determines the level of resistance to departures from controlled flight,

8. Demonstrates proper operation of angle of attack and g-limiter,

9. Determines maneuver limit boundaries if required to prevent unwanted departures from controlled flight,

10. Demonstrates satisfactory flying qualities in the reconfiguration modes, and

11. Demonstrates satisfactory flying qualities in IBU mode.

The specific items evaluated were derived from the system specification and include static stability, stick forces, turn coordination, control harmony, and stall characteristics. A total of 40 flying-quality requirements were identified.

Six general test types were used to evaluate the flying qualities:

1. Preliminary mode checkout consisting of doublet, rolls, and a windup turn.

2. Stability and control tests, including maneuvers mentioned above in item (1), were done at trimmed for level flight and elevated load factor; decoupled control inputs were included. Other tests are mode switching, decoupled control limits, splits, and handling qualities during tracking. Two mission tasks of air-to-surface gun tracking and air-to-surface bombing were performed.

3. High angle-of-attack testing validated the flight control system at high angles of attack, 10° to 30°. Doublets, rolls, and mode switching are some of the maneuvers.

4. A special section was dedicated to testing the flying qualities of the analog back-up unit.

5. The approach of handling dual (complete) loss of an aircraft feedback sensor such as pitch rate, by use of reconfiguring the control laws, was evaluated.

6. The final set of tests evaluated the control system outside the design envelope of a given mode. This test was performed because certain modes, mainly the decoupled modes, had operational envelopes which were smaller than other modes. Inadvertent pilot action could result in operation outside a design envelope.

The results of the flying-qualities test are summarized as follows:

1. The flying qualities of all the flight control modes were satisfactory except for the following items. Because of problems in the advanced modes, only the SNRM was available for the first flight. Any flight tested in other modes would have required retest after the problems were corrected.

2. Flat turn performance was slightly impure, in other words the sideslips were not zero.

3. Pitch maneuvering response was sluggish.

4. Elevator rate-limit cycling occurred in decoupled normal mode.

5. The IBU and all reconfiguration modes except pitch rate reconfiguration mode were accepted for flight test. The fault-tolerant design was sensitive to the pitch rate reconfiguration transition. On occasion, depending on computer skews and aircraft state, the digital system would fail to the IBU during the transition to pitch rate reconfiguration mode. As the IBU needs pitch rate feedback to control the aircraft, this failure sequence would result in loss of the aircraft. This was the only problem corrected before flight test.

5.3.3.4 Failure modes and effects testing (FMET). FMET validates the fault-tolerant design to a level defined by the type of failures induced and the extent to which the system response is measured. To define an optimum set of test cases, FMET requires a functional breakdown of the DFCS and of the software fault-detection system. FMET is performed on the simulator to provide the environment that best approximates flight. In addition to providing essential engineering knowledge, FMET furnishes information on failure effects and interactions, failure transients, and flying qualities of degraded modes, as well as providing pilot experience in fault isolation and the use of emergency procedures.

Because a major objective of the AFTI program was to provide a dual-fail operate capability, FMET required a matrix of single and dual failures. The matrix is composed of first and second failures of the components listed in table 19. A list of each failure type, the failure mode, and its effect is shown in table 20. A matrix of nearly 1000 failure combinations would be necessary to test every dual failure. For this type of testing, a reduction of the failure matrix required careful consideration of the objectives. A primary purpose of FMET was to show that the expected results of a failure are correct; hence any compromise in the failure matrix compromises the ability to ensure safe and proper operation following failures.

FMET was performed in two stages — (1) evaluation test and (2) demonstration test. Evaluation test covered all the tests identified by an "X" in the dual failure matrix (fig. 40). The three additional tests identified in table 19 were added at a later date. An evaluation test was performed by the contractor, and testing did not include government participation. The demonstration test was accomplished using government pilots and engineering support.

Unlike flying qualities, the detail requirements for the fault-tolerant design were not specified in early documents. Though 40 specific flying-qualities requirements were given in the system specification, only one fault-tolerant design requirement was given. It was expressed as (1) dual fail operate and (2) loss of control probability of 1×10^{-7} in a 1-hr flight.

Completing the FMET demonstration required two attempts. The first attempt at the FMET demonstration was not successful because the DFCS's response to failures was incorrect. Part of the problem was determining what the correct system response should be. This was partially owing to the lack of detailed validation requirements. However, the major problem results from the nature of integrated validation testing. This is the first opportunity for the fault-tolerant design, the system architecture (asynchronous computers), and the control law design to operate as one system. This integration testing detected numerous design errors as the separately developed components were required to operate together.

A few of the major discrepancies that were identified during the AFTI F-16 FMET and resulted in modifications to the flight control system included the following:

1. Power failures of a single computer channel resulted in a complete loss of the triplex digital system and transfer to the analog backup;

2. A dual failure of pitch rate information while in power approach resulted in an uncontrollable pitch-down motion;

3. Single failures of an analog-to-digital converter resulted in the loss of the triplex digital system and reversion to the analog backup system.

All totaled, over 40 discrepancies were identified during the 5-mo period in which FMET took place. As can be seen, failure modes and effects analysis is no substitute for failure modes and effects testing. The complex interactions created between hardware elements because of software actions could only be found with the actual hardware and software systems.

5.3.3.5 On-aircraft testing. Several on-aircraft tests were performed to validate that the flight control system operated properly.

Besides the installation and functional tests, three special tests included

1. Structural coupling,

2. Electromagnetic interference-compatibility, and

3. Gunfire tests.

All tests were performed using the airframe with the flight control and avionics systems operating. A brief description and the results of each test follow.

5.3.3.6 Structural coupling. The structural coupling test was performed to show that the response of the flight control sensors to structural vibration does not form a sustained closed-loop oscillation.

A diagram of the test setup is shown in figure 41. Both open-loop and closed-loop tests were performed.

The open-loop test was a frequency response test with the flight control system command to the actuator disconnected. The actuator was driven with an external frequency sweep and the flight control actuator command was then recorded. The input and output commands were then compared and plotted as a function of frequency. A minimum 6-dB gain margin was required.

The closed-loop configuration was tested by inserting a variable gain between the DFCS and the actuators. The gain was slowly increased, with step impulses given to each actuator. The response to each impulse was measured to determine frequency and damping. A 100-percent increase in gain was made for each axis to verify gain margins.

Once a complete DFCS software release was available, a preliminary structural mode test was performed to uncover any flight control-structural coupling. The test identified a pitch axis resonance in the 100 rad/sec range. A correction to the software (control law design) solved the problem using a complex double notch filter. Subsequent testing just prior to first flight found no discrepancies.

5.3.3.7 Electromagnetic interference and compatibility (EMIC). The EMIC test was performed just prior to flight test. The test was extensive, requiring several weeks to complete. All electronic devices were installed and operational for the test. The test ensured that a given electronic component's operation did not adversely affect another's operation. Both interference of, and compatibility between, components were tested. Tests were run using ground power, main generators, and the emergency power unit.

All tests were completed successfully. No changes were required before flight operations.

5.3.3.8 Gunfire test. A requirement to fire the gun during flight test led to a ground gunfire with the flight control system operational. There was a concern that the vibration environment generated by the gunfire, coupled with asynchronous sampling and high-feedback gains, would result in failure declarations. To test the highest gain conditions required an extensive test setup to get the correct flight control mode and air data conditions (Mach number and altitude).

The test results confirmed the fears of the previously described interactions between the gun firing and flight control system. The lateral acceleration feedback path to the rudder command was the culprit causing a computer failure indication after several successful gunfire bursts. The high frequency, large-scale motions of the rudder command differed in the three computers. This difference was owing to errors introduced in the asynchronous sampling of the lateral accelerometers. The three lateral accelerometers and rudder commands, before and after the software fix, are shown in figure 42. A reduction in feedback gain corrected the problem with no effect on flying qualities.

5.3.4 Revalidation of Designs

The types of validation testing are wide ranging and the retesting must be designed to cover as many of the specific areas as possible in each test. Validation retesting must show that corrections to deficiencies do not adversely affect other operations. The seven specific tests performed to revalidate the system are shown in table 21. Test number six is a good example where flying qualities, failure modes, and pilot-vehicle interface are combined in one test.

The revalidation testing was done in addition to tests which specifically validate a given change. The revalidation tests were termed acceptance test procedures. The tests were performed prior to first flight and after each new software release during flight test.

5.4 Qualification Issues

The issues involved with system validation are similar to those of verification. This isn't surprising in that both are testing the implementation of a design, one for the software and one for the system, and both are for an aircraft. The two main issues for system verification and validation are completeness and cost.

Completeness is the primary issue of verification and validation because it has a direct effect on safety and cost. Currently there is a serious lack of standards and tools for achieving and measuring completeness.

In the introduction to the validation section, validation testing was described as testing to see that the system as a whole performs as expected. The completeness issue can be viewed in terms of the completeness of the system specification and how well testing ensures proper operation to that specification. This issue applies to both the software for verification and the system for validation.

The cost associated with verification and validation is the cost of both people and tools to perform the job. The tools needed to perform the test and model the flight environment are costly, especially for validation. The simulation is a primary cost. Secondly, the people cost is high because all the previous activities required to design and implement a change are needed to correct discrepancies. All life-cycle operations must be repeated (see fig. 31), increasing cost. This is a rather well-known and published fact (Brooks, 1979). Two suggestions to reduce the cost of verification and validation are

1. To improve the requirement, design, and test methodology to identify and correct errors as early as possible in the design phase, and

2. To provide an information tool to the designers and testers which can improve the understanding of the design. See section 8 for further discussions on this topic.

6 CONFIGURATION CONTROL

The design of any life-critical system requires a method for resolving discrepancies and controlling system configuration. Configuration control can be defined as knowing what you have and when you have it. A good configuration control process for ensuring safety should contain

1. Visibility of changes across all involved engineering disciplines,

2. Identification of the impacts of a change on requalification, including designed-in testability needed for the changes,

3. Identification of the effects of a change on system performance and limitations, and

4. Identification of the effects of a change on operational characteristics and procedures.

The general flow of the configuration control process is shown in figure 43. Discrepancy reports, which contribute most significantly to configuration changes, are written any time the system does not perform as expected or fails to meet a specification. The cause of the discrepancy is classified as hardware, software, or system failure (hardware and software combination). The report identifies the discrepancy, gives the cause and any possible methods of working around the problem, and details any system limits or operational impacts. The discrepancy report can only be resolved when a satisfactory fix is implemented, documented, and retested.

Changes are controlled through the configuration control board, which provides the forum for disciplinary and flight test engineers to discuss the changes and their impacts, identify retest requirements, and determine the effect on operational procedures.

Aside from discrepancy reports, configuration changes arise from new software, hardware, or system requirements. New requirements enter the change process in much the same manner as discrepancies and follow the same configuration control process. Most of the new requirements on the AFTI F-16 system were classified as improvements. These improvements came in the latter stages of system development and were intended to reduce requirements for computer memory and real time.

7 FLIGHT TEST

Flight test assessed the decoupled flight control, integrated pilot vehicle interface, and to a lesser extent the dual-fail-operate capability of the triplex system. It also provided an assessment of the method and tools used to develop and qualify the DFCS design.

For example, the yaw departure of flight 36 and the resulting data identifying a single failure, which could result in the triplex system failing to the analog backup, are both instances where the qualification methods for the flight control law and fault-tolerant design failed.

It is natural to expect difficulties when pushing the state-of-the-art in a flight research program. However, it is also necessary to understand how the methods can be improved so that the risks involved in future flight research can be minimized.

The results are broken into five sections — General, Fault-Tolerant Design, Control Laws, Software, and Hardware. The results of flight test with regard to the program's goals are summarized in the General section that follows. Considerable published data can be found on this and is included in Ford and others (1984), Ishmael and others (1984), Joyner (1983), and Mackall (1983).

Both ground and flight operations are discussed in the Fault-Tolerant Design section. Lessons learned and a unique flight test discrepancy are summarized in the Control Laws section.

The hardware and software represent the medium in which the fault-tolerant design and control laws are implemented. The Hardware and Software sections will address the failures and errors which occurred in both areas and the implications to system reliability.

7.1 General

Flight test results with regard to the program's goals are summarized in this section. Program goals were to demonstrate and evaluate the following:

1. A dual-fail-operate, triplex digital flight control system,

2. Task-tailored multimode control,

3. Decoupled control,

4. Integration of flight control and avionics, and

5. Pilot vehicle interface improvement of multipurpose displays, head-up display, and voice command.

Flights were successfully tested over a 13-month period in 1982 and 1983 (fig. 44). Pilots from NASA, U.S. Air Force, U.S. Navy, and General Dynamics flew a total of 118 flights. All major objectives were completed, including envelope expansion for high angles of attack, flight at Mach numbers up to 1.2, combat mission evaluations of decoupled control, and structural load clearance for the decoupled modes. Low-flight rates early in the program were owing to anomalies of the basic aircraft as well as to the AFTI unique systems. Thirteen software releases were made to the digital flight control system during flight test. Software changes were made to correct discrepancies and to provide improvements in flying qualities, fault-tolerant operation, and structural-load-limit items. An efficient software change process was required to provide safe, timely changes needed to meet flight test objectives.

The specification for the probability of loss of control for the AFTI F-16 was 1×10^{-7} per flight hr. This specification addressed only hardware reliability, not software reliability, and assumed accurate detection of failures. The fact that there were no confirmed flight control computer hardware failures indicated an excellent reliability rate based on the number of flights completed.

Avionics and the flight control system were integrated satisfactorily. However, one failure occurred to a critical nonredundant avionics system which adversely affected DFCS operation (see section 7.2).

Built-in test is a highly automated test sequence that ensures the digital flight control system is free of hardware failures prior to takeoff. The BIT is run prior to each flight and takes approximately 2.5 min. Two failures of the hardware were detected by BIT during preflight testing. The first was a failure of a surface actuator, and the second involved memory chips which didn't meet timing specifications at cooler temperatures. Nuisance failures of BIT occurred a number of times. The cause was believed to be due to electromagnetic interference.

Faults were detected in-flight by comparing the three channels' values for tracking across the different channels. The only real failure was an input signal which was traced to a pushed-back pin in the aircraft wiring. Fifteen false failures occurred owing to design deficiencies rather than actual hardware failures (table 22). The design deficiencies caused both temporary (resettable) and permanent loss of flight control redundancy. These design deficiencies resulted from the coupling of unique computer skews with characteristics of the flight environment, such as sensor noise. Undetected during qualification, these in-flight failures resulted in envelope and flight control mode limitations until they were corrected by software changes.

The asynchronous computer architecture affected a wide range of developmental activities including design, software-system qualification, and flight test operations. The DFCS qualification was complicated by the dependence of failure modes on computer skew. Testing at predetermined worst case computer skew improved testing results; however, some deficiencies still escaped detection. Ground operations during aircraft preflight were also impacted by the asynchronous computer architecture. The most common problem was DFCS failure, requiring reset by the pilot or cycling of aircraft electrical power.

The 13 flight test software releases, in which design, coding, and test were performed at General Dynamics, Fort Worth, supported the needed changes of flight test. The first four releases provided full envelope capability for the AFTI vehicle in all flight control modes. The remaining nine releases modified the control laws to improve flying qualities and provided corrections to the fault-detection design deficiencies.

Although the IBU was never engaged as a result of a digital system failure, flight test experience indicates that IBUs are needed. The complexity of the IBU became a primary issue; a simple IBU could not provide protection at envelope extremes which are possible to reach with the primary digital system. Furthermore, the relaxed static stability characteristic requires a certain level of augmentation. The simplified reversion mode used on AFTI provided get-home capability and level two flying qualities for landing as specified. However, simulation and flight test indicated a more capable IBU is needed to cover transitions at the envelope extremes possible to reach with the digital control system (Ishmael and others, 1984).

The flight test results for the control laws and flying qualities are summarized in the following paragraphs with additional information available in Ford and others (1984).

A primary objective of the AFTI F-16 program was the evaluation of a multimode, task-tailored digital flight control system with decoupled aircraft control. The six different decoupled options and right-hand control options were evaluated with the decoupled feature best suited for a given task being identified. The task-tailored approach provided improved handling qualities for the tasks evaluated.

The AFTI F-16 control laws that showed the most improvement relative to the production F-16 were all pitch rate command systems. In all cases this control law structure was demonstrated to have good open-loop stability characteristics, good dynamic response characteristics, and an attitude hold feature (auto-trim) that reduced pilot workload.

The adaptive control law, which used pitch rate error to optimize performance for gross acquisition and fine tracking, was shown to be the best option for the air-to-air combat task. The adaptive gain control law was implemented using the right-hand controller; decoupled pointing with the pedals and twist grip showed no significant improvement for the air-to-air task.

The best features for the air-to-ground task were improved flight path stability and ride smoothness in turbulence in the pitch axis. Direct side-force or flat turn, which is commanded through the rudder pedals, improved the task and reduced pilot workload for obtaining lateral axis solutions.

Problems with roll ratcheting affected all the modes except standard normal. Prefilter tuning was not sufficient to resolve completely the ratcheting problem.

The standard normal mode improved the pilot's workload for the power approach task. Using more of a pitch rate command system rather than the normal acceleration command system on the production F-16s, improvements in flight path and angle-of-attack stability were made.

7.2 Fault-Tolerant Design

The in-flight, preflight BIT, and ground test anomalies of the fault-tolerant design are discussed in this section.

7.2.1 In-Flight Experience

The flight test results for the fault-tolerant aspects of the DFCS are summarized in table 22. The cause "system integration" indicates that the failure was caused by a design oversight which was discovered when separately designed systems were required to work together. For example, in most of the cases where the correction was to vote software switches, the cause was owing to the integration of asynchronous computers, control laws, and the fault-tolerant design.

The most critical anomalies occurred on flights 15 and 44 and are summarized below. The anomaly involving a roll axis software switch is also discussed. A design oversite in the electrical system resulting in flight control operation on batteries concludes this section.

7.2.1.1 Anomaly of flight 15. The stores management system (SMS) sends pilot requests for mode changes to the DFCS via the avionics multiplex bus (fig. 13). A failure of the SMS — it is not known whether in the hardware or software — resulted in DFCS mode change requests at 50 times per sec. The DFCS responded at a rate of 5 mode changes per sec. The pilot was not maneuvering at the time of the failure.

The rapid mode changes were identified in the ground control room and the SMS was powered off by the pilot, stopping the mode changes. The pilot commented that the aircraft felt like it was in rough air, owing to the different surface trim positions corresponding to the various flight control modes. The flight was aborted and the aircraft landed safely.

Analysis of the anomaly was conducted using the DFCS hardware in the loop simulation. Results of the investigation indicated that if the pilot had been manuvering, a complete failure of the DFCS to the analog back-up would have occurred. Maneuvering would increase the difference between surface positions for the control modes. The difference would be interpreted as a failure.

Flight test continued after a software modification was made to improve the DFCS's immunity to this failure mode.

7.2.1.2 Anomaly of flight 44. Prior to flight 44, three occurrences of failure indications for a single branch had occurred (flights 23 and 28 of table 22). Concerns that random computer skew between the three computers would lead to multiple channel failures had been accepted as an allowable risk. On flight 44, a control law software switching mechanization coupled with unique flight conditions to produce a divergence of output commands in the three computer channels. The divergence resulted in what appeared to be dual, simultaneous computer failures. The failures were caused by slightly different air data valves in the three channels initializing integrators with different values. This caused output command divergence between channels. The fault detection logic was such that each channel of the DFCS declared the other two channels as failed. In this situation the design was supposed to result in the analog back-up mode, however this did not occur. Another design oversight in the redundancy management software kept the analog back-up from being selected automatically. Dual simultaneous failures had been ruled out as not possible, therefore the design did not account for them. The system could not be reset by the pilot, even though no actual hardware failure had occurred. The aircraft was safely landed with only one of the DFCS channels controlling the aircraft.

7.2.1.3 Anomaly of the roll axis software switch. Another example from flight test illustrates how the asynchronous system design and the lack of modeling sensor noise during the testing phase can affect flight test operations. A failure indication in flight was traced to a software switch in the roll axis command path. The software switches controlled the paths through the control laws. If a switch was to change condition in one channel and not the others, an output miscompare would be detected and perceived as a hardware failure. A schematic of the software switch, a function of note N, is shown in figure 45. The note N logic conditions are fairly complex and are not shown. Note N logic controlled a forward integrator used for steady state decoupling and was based on the size of roll stick and rudder pedal commands.

To correct the problem, a software change was made. The software switch action was voted to ensure that all channels had the same switch position and control paths. Extensive simulation testing was performed to show that the voting of the switch kept the control paths the same in all channels. All software coding and simulation testing was passed successfully, indicating that the voting of the switch action was correct.

The first flight test attempt to repeat the test point, which had induced this failure, resulted in another failure indication. Analysis of the repeat anomaly found that although the switch action was voted, the old, unvoted value was still being used to control switch position. An error in the software coding had occurred and passed through the verification and validation testing. This was the only case of a software error being found in flight.

When the exact conditions which identified the design oversight were known, the random nature of asynchronous operation, coupled with lack of modeling for sensor noise, allowed the error to pass testing undetected. This example graphically shows why the failure indications of table 22 occurred. It is easy to see why some design errors passed through testing and were only found during flight test. Prior to a failure, the exact flight conditions and aircraft configuration which uncover a design flaw are not known.

7.2.1.4 The electrical system anomaly. This in-flight anomaly gives an example of a design oversight which surfaced in the electrical system. The in-flight anomaly caused the flight control system to operate off battery power, although no actual electrical system failure had occurred. The mission was aborted with the aircraft landing safely.

As discussed in section 4.1.6, the AFTI electrical system included overvoltage detection relays to protect the flight control system from overvoltage failures of the emergency generator. Figure 18 shows the electrical system schematic. The overvoltage detectors monitor both essential dc busses and the output of the emergency converter. The emergency converter provides power to the flight control system in case of power loss from (1) the main generator, (2) the emergency generator, and (3) both batteries. The emergency converter derives its power from the permanent magnet generator (PMG), a portion of the emergency generator. The PMG will provide a small amount of power for certain classes of emergency generator failures. The PMG-emergency converter would only be needed in the rare case when all other power sources had failed.

In-flight the pilot was practicing a simulated engine flame-out maneuver which calls for turning on the emergency power unit (EPU) that drives the emergency generator and PMG (fig. 46). When the EPU was energized with the high engine rpm, the energy surge caused the PMG voltage through the emergency converter to peak at 36 V. Previous ground test had been performed at lower engine rpm. The overvoltage detectors, set at 35 V, disconnected the emergency converter and the essential dc busses from the flight control system. With the protection relays open, the unloaded PMG voltage through the emergency converter went to 40 V, keeping the relays latched. The flight control system was being powered by the aircraft batteries.

Ground test with engine rpm that matched the flight condition confirmed the cause of the anomaly. The interactions of the EPU, emergency generator's PMG, and the overvoltage protection relays caused primary electrical system disconnect. Although all the components were operating within their specifications, it required the unique condition of EPU operation at high engine rpm to reveal the design oversight.

7.2.2 Ground Experience

Problems encountered during ground operations were associated with two separate operating modes. First is the BIT operation, and second is the memory mode operation. The BIT and memory mode operations are described in section 4.1.8.

7.2.2.1 Built-in tests. The BIT operational experiences are summarized in tables 23, 24, and 25.

The BIT detected four actual hardware faults — two where solid-state components, one a relay, and one actuator failure. The most elusive failure was the solid-state random-access memory (RAM), which failed at temperatures below 40°F. Once the problem was identified, all memory chips were screened and several replaced.

One software error was detected by BIT. The error was in the BIT software itself. The BIT would not pass on the aircraft and therefore that software release was never flown. The error was the result of simple oversight. An improvement of BIT to test for parity errors as a final check detected the parity error it had purposely set in a previous test.

Numerous bit failures occurred for which the reason was never resolved (table 24). The majority of the failures were believed due to EMIC, however this was never confirmed.

The BIT failures owing to system integration weie the most interesting (table 25). In these cases a lack of understanding of the system operation, at times combined with incorrect procedures, induced BIT failures.

Item number 2 (table 25) provides a good example of how the integration of systems caused problems with BIT. In this case BIT failed a test of the avionics multiplex bus, and the communication from the multipurpose displays to the DFCS was lost. This caused a lockup of the system which required cycling of DFCS power to correct. The problem was caused by one side of the dual SMS failing to pass information between the DFCS and the MPDs. The BIT design did not allow for switching to the other side of the communication bus for this failure as the in-flight software does.

It can be seen by comparing the number of actual faults detected, five, to the number of nonrepeatable and system integration faults that EMI, system complexity, and the resulting lack of detailed insight caused as many problems as the hardware itself.

7.2.2.2 Memory Mode. The memory mode option was an excellent test tool for determining DFCS state. It was very valuable when troubleshooting system failures. When in memory mode, only available on the ground, all normal DFCS flight computations are halted. Two unique anomalies occurred with memory mode operations (table 26). Both cases are examples of system integration problems. The first was a failure of a DFCS channel when entering memory mode with rudder pedal input. It occurred during taxi when the pilot was using the rudder pedals for nose wheel steering. The asynchronous operation of the computers resulted in entering and exiting the memory mode at different times in the different computers. With dynamic pilot inputs, memory mode selection would cause DFCS channels to believe they had failed.

The second memory mode anomaly occurred when the engine was started with the control system in the memory mode, which does not have the power-up and restart routine of the in-flight software. The power transfer, caused by the aircraft generator coming on at engine start, failed the DFCS computers. Unfortunately the failure mode resulted in a canard hardover, damaging the nose gear door.

7.2.3 Summary

The criticality and number of anomalies discovered in flight and ground tests owing to design oversights are more significant than those anomalies caused by actual hardware failures or software errors. The two design oversights discussed above, and identified as being caused by a lack of understanding of the system or a system integration problem, can only be avoided in the system qualification and design life cycle phases.

Although the failure indications were of computer hardware, testing of the hardware alone will not find the error because the failure indication is declared by the software. Testing of the software alone would not detect the error because the software was implemented correctly, per its specification.

Only during system qualification when the hardware and software system are operating in an environment which is nearly equivalent to the flight environment can design flaws such as these be found and corrected.

Qualification of such a complex system as this, to some given level of reliability, is difficult for two reasons. First, there is no established method of identifing system level requirements and relating them to the needed system level testing. Secondly, as discussed in section 5.3.3.4 on system qualification, the number of test conditions becomes so large that conventional testing methods would require a decade for completion. The fault-tolerant design can also affect overall system reliability by being made too complex and by adding characteristics which are random in nature, creating an untestable design.

As the operational requirements of avionics systems increase, complexity increases. Reducing complexity appears to be more of an art than a science and requires an experience base not yet available. If the complexity is required, a method to make system designs more understandable, more visible, is needed.

The asynchronous design of the triplex DFCS introduced a random, unpredictable characteristic into the system. The system became untestable in that testing for each of the possible time relationships between the computers was impossible. This random time relationship was a major contributor to the flight test anomalies. Adversely affecting testability and having only postulated benefits, asynchronous operation of the DFCS demonstrated the need to avoid random, unpredictable, and uncompensated design characteristics.

7.3 Control Laws

As described previously, the task-tailored control mode options provide a uniquely tuned control law for a given task. Designing the control mode for a specific task instead of one general control mode for all tasks improved the aircraft's performance. The following discussions will address the most interesting anomaly involving the control laws.

The yaw departure on flight 36 was the most significant control law anomaly. A review will provide insight into its cause. The yaw departure occurred in the SNRM mode during a maximum rudder step and hold input. Mission rules limited the maximum sideslip to 10°. Practicing this sideslip maneuver on the simulation showed that the 10° limit would not be exceeded.

In flight test, the maneuver resulted in a temporary sideslip excursion to 14°; from there a rapid departure from controlled flight occurred. The aircraft departure was of a short duration, approximately 3 sec, but resulted in some extreme conditions and flight control system failure indications.

During the departure the sideslip exceeded 20° and normal acceleration exceeded -4 g, then +7 g (fig. 47). The aircraft rolled 360°, angle of attack went to -10° then to +20°, and all control surfaces were operating at rate limits. The departure was quite severe aerodynamically, resulting in the vertical tail exceeding its design load limits. Flight control system failure indications included hydraulic system failure for both canard actuators and an air data failure. The failures were transient and were reset after control was regained by the pilot.

After analyzing the problem with the simulation, the following reason for the departure and its subsequent failure indications was found. The aerodynamic model used to develop the control laws and used in the real-time simulator was insufficient for the conditions of the discrepancy. The lateral directional derivatives were a function of sideslip, but only modeled to ±10°. Secondly, the nonlinear nature of the derivatives as a function of sideslip was not modeled. The wind tunnel data were modeled as a straight-line function, giving the simulation more restoring force than the aircraft.

The problem was corrected by removing the canards from the command path so that the aircraft could not obtain 10°

of sideslip. The horizontal tail structure was examined and found to be undamaged, and a structural modification was made to the vertical tail to increase its load limit for future flights.

The hydraulic vote in the canard actuators was owing to a drop in hydraulic pressure as a result of all control surfaces being at the command rate limits. The air data failure, although appearing to be straightforward, proved to be quite interesting. The air data failure was a transient failure caused by the side-mounted probe which was blanked by the fuselage at the high-sideslip angles. A detailed review of the three computers' surface commands showed a mistracking during this failure. Analysis showed the S/M technique passed the side probes error through until the failure threshold was reached (fig. 48). The air data information is used to determine flight control law gains asynchronously at 4 times per sec. The air data failure transient, shown in figure 49, caused changes to the control law gains, giving different control surface commands in the three channels.

Fortunately, at the flight condition of the departure, the gain changes did not produce differences which would cause failure declarations of the computers. For several areas of the flight envelope, particularly at high angles of attack, this single air data failure would result in failure of the DFCS to the analog backup. This increased risk was accepted until the software was modified.

Several points should be noted from this flight incident:

1. Any simplification in the modeling of the aerodynamics must ensure that it is conservative with respect to its effects on the aircraft as a whole.

2. The aircraft must be considered a system consisting of highly related disciplines and functions. The control law design error caused by the modeling error revealed design errors in the hydraulic system and DFCS fault detection logic. The aircraft structure became involved due to control law design error.

3. To thoroughly qualify an aircraft with these types of systems, one must

A. model dissimilar sensors completely, including sideslip effects;

B. test with the aircraft in the loop with the simulation, thereby including the actuation system;

C. test failure modes other than hardover failures; and

D. have a complete understanding of the DFCS design and its interrelationships.

7.4 Hardware

The DFCS hardware includes the F-16 baseline sensors and controllers and the AFTI flight control computers and actuator interface unit. Based on repeatable, isolated failures, the reliability of the hardware was excellent. In the 175 hr of in-flight test, an electrical connector was the only hardware failure. In the 6200 hr of ground time, including time prior to flight test, only three failures were documented in the computers. To determine computer reliability, the time must be multiplied by three, for the three computers used in the system. This gives a 6200-hr mean-time-between failures compared to a predicted 1200 hr.

The most significant problem to address lies in the number of nonrepeatable failures and failure indications listed in table 22 and discussed in section 7.2. Depending on one's operating rules, these failures can result in considerable equipment changes and loss of aircraft availability.

7.5 Software

The use of a DFCS enables changes to the systems' characteristics, such as flying qualities, by reprogramming the software. Efficient and safe flight test requires a thorough method for evaluating, implementing, and testing software changes.

In the 1 yr of flight testing, 129 software changes were made in 13 separate releases (fig. 50). A software release is a package of changes provided in one update.

The process of changing the software is described in section 6. The majority of the evaluation, implementation, and testing occurred at the contractor's facility. The Joint Test Force, consisting of the Air Force Flight Test Center and NASA, played two major roles in the software process. First, it identified changes that were needed and their timing for release, and second it provided an independent audit of the changes. The audit included independent validation testing at the contractor's facility and review of supporting software products, such as the documentation.

Three errors were found in the software releases after they had been approved for flight test. All resulted from changes, and none were latent errors which existed for several releases. Two of the errors were found in ground preflight tests, and one was detected during flight.

A software error is defined as the software not operating in accordance with its specification.

The first error in the software resulted from a change to the BIT software. The change was to cause BIT to fail if it detected a parity error in the hardware. One function of BIT was to read a memory location with known bad parity and to check for parity error detection by the hardware. The software used to detect unintentional parity errors also detected the purposeful one, causing the BIT to fail erroneously.

The second software error involved the structural limit for the vertical tail. The structural load was limited by restricting the rudder command as a function of impact pressure and flight control mode (fig. 51). During flight test, it was found that the vertical tail loads were exceeding those desired with the rudder limit implemented. Several options to correct the problem were engineered and evaluated with the hardware-in-the-loop simulation. The best option was identified and included in the next flight release. Unfortunately, one of the other options being evaluated was accidentally left in the software. The error was found during ground testing which checked the installation of the new software into the aircraft. Tests showed that the rudder was being limited to smaller deflections than those expected. The software error gave a conservative limit for vertical tail loads, but resulted in unnecessary operational limits. A decision was made to use the software release until it could be corrected in the next update.

The last software error was discussed in section 7.2. It involved a change to the software, voting a software switch, causing all three computers simultaneously to use the same control law paths. Although extensively tested prior to flight with software unit tests and hardware-in-the-loop tests, the anomaly was not detected until flight test. The software error was one of correctly voting the three computers' determination to toggle a software switch, but then not using that voted value for the actual switching action. The previous method was still in effect, that is, each computer channel determined its own switching. Flight tests under the conditions that would allow this error to occur, which were a function of control mode, were prohibited until the next software release.

8 OBSERVATIONS AND RECOMMENDATIONS

Each of the four life-cycle phases are pulled together in this section to examine how they affect one another. The approach in this section is to first give a detailed case study, then to summarize each of the development phases. Emphasis will be on how the flight test phase was affected by the previous three phases — specification, design, and test. Comments are also given which only refer to a single life-cycle phase, such as recommendations that would provide for a more efficient qualification. After completing the case study, a review of the previous three development phases gives three perspectives on how anomalies can be avoided and how to maximize the benefits of flight testing. Looking at system-software qualification, we see that more complete and efficient testing is needed. Looking at design, we see that operational benefits can be achieved by improving system architecture. Finally, when considering the system specification, we see that if sufficient detail exists to identify exactly what is desired, then a correct design is more likely. The case study from flight test will help to clarify each of these concepts. Following the case study, recommendations for each life cycle are summarized.

8.1 Anomaly of Flight 44, A Case Study

The anomaly of flight 44, discussed in section 7, provides a good example to illustrate how activities of the previous life-cycle phases contributed to a flight-test anomaly. The anomaly was owing to a design oversight and required several unique conditions, which are outlined as follows (fig. 52):

1. Standard combat or a decoupled flight control mode had to be active.

2. The pilot had to have full rudder pedals, flying at 170 knots calibrated airspeed (KCAS).

3. Sensor noise coupled with computer skew had to give a 3-knot difference in the impact pressure values in the three computer channels.

In the flight control modes identified, a rudder fader, schedule D69, removes pilot commands below 170 KCAS, for controllability reasons. The difference in the perceived airspeed for the three different channels allowed different amounts of the full-rudder-pedal commands to pass through schedule D69. The three different pedal commands initialized each channel's integrator, resulting in a divergence of the output commands to the canard surfaces. Each of the three computer channels declared the other two as failed. The aircraft was landed effectively with a single string flight control system.

8.1.1 Specification

Clearly, it is not desirable to have a system design that can cause loss of system redundancy when no failure exists. However, there is nothing in the specifications which addresses incorrect failure detection.

Reliability requirements simply address component failures. Once you specify that incorrect failure detection is not acceptable, criteria are needed to ensure it. This has been a matter of engineering judgment to date.

Recommendation

Incorrect fault detection, resulting in inappropriate loss of system redundancy, is unacceptable. The criteria for ensuring proper operation should be to test voted-compared values with inputs that represent the physical limits of the device-system in question. The physical limits to consider include rate of change, minimum and maximum values, maximum frequency response, and noise, as examples.

8.1.2 Design

The design change needed to avoid this anomaly is one ensuring that the same value of a sensor is used in all redundant channels. Simultaneous sensor sampling and proper sensor selection routines would ensure congruent sensor values.

Recommendation

Redundant system designs which use voting and cross-channel comparisons to detect faults must operate on congruent input data sets to avoid incorrect failure detection.

8.1.3 Qualification

For the system-software qualification activity, several generalized techniques should be used to detect the anomaly of flight 44 prior to flight test. Testing the control mode condition for the anomaly is easily done, in fact these control modes were tested for months using the hot bench simulation. The rudder fader, likewise, was tested numerous times.

However, the amount of impact pressure error in the different channels was never enough to cause the problem to appear during ground testing. Sensor noise and computer skew were two parameters which were not controlled, nor were the exact triplex values known during the ground testing.

Recommendations

1. Fault-tolerant system design must be evaluated for sensitivity to sensor noise.

2. A computerized description of the flight control system is needed to identify conditions, control modes, and flight conditions for doing sensor noise sensitivity tests. The computerized system description would accept user inputs, such as flight conditions and control modes, and return active command paths appropriate for sensor sensitivity tests.

3. Random unmeasured system parameters such as computer skew must be limited. If they can't be limited, additional testing is needed to get a statistical base for predicting its effect on system operations.

8.2 Observations and Recommendations by Development Phase

To minimize the detailed discussions needed to review every flight anomaly and the analysis used to arrive at each recommendation, we will briefly describe the observations and recommendations as they apply to each development phase.

8.2.1. Specification

The primary observation concerning the specifications is the lack of detailed specifications for reliability and fault tolerance. The majority of the specifications is concerned with stability and control requirements for conventional, nondecoupled control system designs.

Recommendations

1. In addition to the loss of control and abort specifications, failure probabilities should be given for the different mission phases and the functions performed by the flight control system. By specifying abort probabilities for different missions, such as air-to-air intercept and air-to-surface bombing, the designer can avoid either over- or underdesigning the system. The reliability of functions, such as pilot displays and controls, should likewise be given reliability values.

2. Reliability requirements need to address the software by identifying the testing methods and tools to be used and by clearly stating the requirements of any independent backup, whether hardware or software in nature. The key to software reliability is not found in a failure rate, but in the examination of the method and tools used to ensure proper functionality. The software's life cycle of specification, design, and test must be specified so that testing is traceable to the requirements, and proper functionality is shown. The appendix provides some detailed testing examples for control law functions.

Requirements for an independent backup should include (1) method for detecting the need for a transition to the backup, whether manual or automatic, (2) allowable transition periods and transients, and (3) functional requirements of backup, such as operating envelope and reliability. If the backup is going to be flight tested, reengagement of the primary system must be addressed.

3. Failure transients should be specified in terms of the resulting aerodynamic and structural effects. Table 27 is an example of maximum aerodynamic failure transient requirements, which vary with mission. At certain conditions, such as high-impact pressures, a surface transient can result in structural damage with little aerodynamic transient. Table 28 shows a possible specification for structural transients by giving flight condition and surface transient allowable, thereby implying flight loads. The conditions would be derived from calculations which determine excess structural loads.

8.2.2. Design

Some observations and generic recommendations for designing fault-tolerant control systems are presented in this section. The major theme of the recommendations is based on the life criticality of the control system and handling the complexity imposed by redundant systems.

Reviewing the methods used to develop the system architecture, software, and control law designs shows that both the method used to specify the control laws and the tools available to develop them are more mature than for the other two. Whereas, some software tools are available for the software development process, tools to assist in specifying and performing tradeoff studies are needed. Tools do not exist for the system architecture and fault-tolerant design task.

There is no integration of tools for the three disciplines. For example, DIGIKON, used to develop the control laws, has a data base which describes the control laws. DIGIKON is not tied to any of the software development tools. A laborious handmade description of the control law design had to be written for the software mechanization document.

If a system made of these three elements is to work as a whole, development and integration of design and development tools are needed.

Recommendations

1. An integrated design tool, which addresses control laws, fault tolerance, hardware, and software, is needed for fault-tolerant control systems. A few of the capabilities needed in such a tool include the following:

A. Documentation of the system design in a computer data base which relates the different functional areas. The data base would be queried to find possible interactions, such as sensor noise, affecting command paths in the control laws.

B. Evaluation of the system design for fault tolerance, control laws, and software execution prior to actual system build. The ability to analyze the design and make corrections prior to building hardware and software code would reduce redesign during qualification.

2. When designing interfaces to a redundant flight control system, one must carefully consider the criticality of the information being passed and the failure modes that are possible. This requires a detailed understanding of the items being interfaced. A case in point was the ISA and flight control system interface. Additional testing of the ISA was needed to design its interface.

3. The avionics interface is an example where no redundancy existed for many of the failure modes possible. The information passed from the avionics was critical enough to have caused a failure of the DFCS. The redundancy required in an interface must be based on the criticality of the information and the possible failure modes.

4. A fault-tolerant system, which uses cross-channel voting to detect failures, should avoid random, unmeasurable design characteristics, such as asynchronous channel operation. This helps to keep failure thresholds at low levels and minimizes unexpected interactions that can result from incongruent data sets.

The fault-tolerant design should also be transparent to the control law functions. The control laws should not have to be tailored to the system's redundancy level.

8.2.3 Qualification

The leading issue in qualifying or testing complex DFCSs are completeness and cost. Test completeness is an issue with any software-driven system, but becomes a major item when the system has full authority control of a piloted aircraft. Determining some level of test completeness is also difficult because of the complexity — number of dependent inputs and number of operating modes. The appendix provides some suggestions for complete testing of control laws.

Cost, the other leading issue, results directly from the effort needed to completely test complex systems to a reasonable level. Rather than achieving a measurable level of test completeness or by meeting established criteria, the amount of testing performed often becomes limited by cost.

Recommendations

1. System-software qualification testing must be performed to ensure implementation of the requirements, and that each requirement is tested to meet an established criterion. For example, a system requirement for decoupled control would result in a corresponding

software requirement, identifying specific control law components, such as those given in the appendix.

Testing must be identified for both the system and the detailed software requirements. The testing must be complete enough to verify the requirements are met at both levels. A tool or method is needed to ensure test coverage of all software components.

2. Tools that support automation of the verification task should be used. Automated test stimulus, data recording, and analysis can provide for more thorough tests, better test documentation, and more efficient use of personnel. The use of qualitative pass-fail criteria, such as reasonable transients and acceptable differences, should be avoided.

Test automation will require real-time instrumentation of internal software calculations. The system design will need to support the special test requirements for providing visibility into the system.

3. A computerized on-line description of the system (see recommendation 1, section 8.2.2) should be available to test engineers. This data base of design information will assist the tester in determining test conditions, functional interactions, and parameters to record for assessing proper test results. The data base would also be valuable for determining the cause of discrepancies.

4. System testing must consider the operating environment in which it is going to be used. Vibration and temperature effects on sensor values used by the control system must be modeled. These effects can easily be implemented by imposing biases and noise on the simulated sensor values.

8.2.4 Flight Test

By far, the best thing that can happen to a flight test program is to have thorough specifications, a good design, and a complete and efficient qualification. The fact that more anomalies and flight test time were lost owing to design oversights than actual component failures attests to the need for improving the development cycles.

Some specific recommendations from flight test follow.

Recommendations

1. To ensure the best system configuration for retesting of changes and corrections during flight test and to minimize downtime to resolve flight anomalies, it is recommended that the aircraft design include the capability of closing the aerodynamic loop around the aircraft with the flight avionics installed. This configuration minimizes the number of unknowns involved when testing. Unexpected interactions which have not been modeled can be detected.

2. A computerized data base describing the aircraft's flight system would greatly help flight test. Used as an educational tool for new engineers, it could reduce the learning curve. As during the system qualification, it would be valuable for troubleshooting flight test discrepancies.

3. Increased visibility into the digital system requires instrumentation of intermediate software calculations. To effectively analyze system performance and resolve anomalies, data from internal calculations are required at the frame rate they are being calculated. The aircraft flight instrumentation system and postflight analysis systems will need to support the increased data flow imposed by this requirement.

9 CONCLUDING REMARKS

The flight test program on the AFTI F-16 validated the concepts of decoupled flight control and the integration of avionics functions in the cockpit to reduce pilot workload. Just as important, it provided a chance to evaluate the tools and methods used in its development. The performance capabilities demonstrated by the AFTI F-16 required a new, higher level of avionics complexity. Flight testing provided the environment and conditions to uncover the design advantages and oversights.

To minimize the oversights that came from working at the leading edge of technology, recommendations are given to improve all the development phases. For the specification phase, allowable failure transients are presented which specify the aircraft motion and structural loads permitted owing to a flight control system failure.

The asynchronous digital control system design is reviewed and the problems of using this approach examined. Creating a computerized description of the system design is proposed to help evaluate designs prior to committing to build.

For the testing phase, several recommendations are given to help reduce cost and flight test risk. Automated software testing is one approach proposed.

The benefits shown during flight test of the decoupled control modes are presented, showing the advantages of commanding direct sideforce. The anomalies discovered in flight testing are explained in detail, with reflections on how they might have been avoided.

Overall, the integrated digital control system provided many operational benefits. The hardware reliability of the complex system was excellent. However, the complexity of the system, coupled with the wide range of disciplinary engineers involved, caused numerous design oversights.

National Aeronautics and
Space Administration
Ames Research Center
Dryden Flight Research Facility
Edwards, California, January 13, 1986

APPENDIX — VERIFICATION REQUIREMENTS CONTROL LAW

One required level which flight-critical control laws must be tested to is specified in this appendix. This testing is performed in the actual hardware environment with all software operating to show that the control laws operate properly with, and in the presence of, all other software routines. This method for testing complex control law software is based on a divide and conquer philosophy.

The control laws are broken down into individual blocks for which there is one input and one output. These individual blocks are tested, and interconnections between the blocks are checked. Static checks should be done first, followed by the required dynamic tests. After the lower levels are tested, end-to-end checks for comparison to equivalent FORTRAN-implemented control laws are done. The following will address the method in which lower levels can be tested in the actual hardware environment with all software present.

Step 1

Break down the control laws into individual blocks. Figures 53 and 54 are lateral-directional control law diagrams. One section of this diagram has been broken down into modules and is shown in figure 55. This breakdown needs to be refined to provide functions with one input and one output. Figure 56 shows the breakdown to individual blocks. Blocks can be combined for testing provided proper implementation can still be shown.

Step 2

Identify types of functional blocks to be tested and the type of tests required. A list of dynamic and special tests which need to be performed for each function follows.

Variable Gains Scheduled on Air Data and Other Parameters

Sweep through the full range of the scheduling parameters while recording the gain values. An input against output cross-plot routine will provide data for comparison to the specification.

Fixed Gain

Modify the gain and rerun the static check. This checks the gain's position in the control law loop and proper scaling effect.

Dynamic Elements, Filters, Integrators

Step inputs are applied to the input with resulting output time history responses recorded. Comparisons to independently implemented elements are made to show identical time history responses.

Nonlinear Elements, Stick Shaping, Limiters, Deadband

These elements require full-range input sweeps with outputs recorded. Cross-plots of input against output can be compared to design data.

Multipliers

Multipliers are checked in the static checks; full-range positive and negative values should be checked. Proper system response to overflow conditions must be tested.

Summing Junctions

Summing junctions are also checked in static tests; full-range positive and negative inputs are required.

Switching Functions

The switch connection and the functions which cause the switching action are tested. Do not attempt to toggle the switch by fooling a memory location; set up actual input conditions which cause the switching action.

Block Interconnects

In the static checks, the output of each block is checked for proper connection to other block inputs.

Scheduled Dynamic Elements, Filters Scheduled on Air Data

This type of software function is impossible to test completely. The scheduled value must be tested like the variable gains. Several values must be chosen for the variable with step responses measured. Worst case and extreme values should be used.

Rate Checks, Rate Limiters

Rate checks must be tested just below and just above the rate check level. The signal should be passed unaltered below the level. Above the rate check or limit, flagging or limiting should occur.

Other Possible Digital Functions — Delays and Decrements

Any other unique functions must be examined and proper static and dynamic tests determined to show correct implementation. Emphasis should be on worst case and extreme values as well as showing proper implementation.

Step 3

Determine design requirements and modifications required to test the software.

1. Inputs and outputs of blocks must be made accessible for external recording and plotting by storing these intermediate values in memory for output to a recorder.

2. In order to carry out dynamic tests of internal blocks, a test program to produce a step input is required. This function needs

   (a) Step size and duration,

   (b) Input location of step, and

   (c) External method of starting step function.

This step program can be patched in for software testing and then be disabled for flight. With this function and by disabling the store instruction for the output of the previous block, dynamic tests can be performed.

3. A general purpose digital-to-analog converter output program is useful and would allow putting out any memory location on a spare digital-to-analog converter (DAC) channel. Information needed by the program includes

   (a) Location for output,

   (b) Scale factor,

   (c) Bias correction, and

   (d) DAC channel for output.

This program can also be a test patch. If a test patch is not used and this software will remain for flight, proper lockouts must be included and verified.

Step 4

Since time on the hardware system is usually at a premium, post-test analysis of data is needed. Post-test anlysis requires that recordings of data be made and that a method for plotting the data be available. Cross-plots and time history plots are both needed (see tests for functions in step 2). Recordings of digital data from the computer's memory provide the best flexibility.

Digital control laws are often dependent on external conditions, such as landing gear up or down or a given angle of attack for the alpha limiter. When testing control law functions, these external input conditions should be set by placing the conditions on the input analog and digital signals. In this way, the software system interactions can be tested. Falsely setting internal flags will not allow control law software to interact with the rest of the software structure.

The use of the simulation in a static mode, or by adding some special capabilities into the simulation, can provide the necessary input conditions. Since the simulation has all the inputs required to drive the control laws, test conditions can be set easily by using a cathode ray tube terminal tied to the simulator.

Special simulator capabilities to augment testing could include

1. Ramp function, that is, sweep alpha from -5° to +50° in 10 sec,

2. Step and sine functions, and

3. Predetermined logic or flight conditions.

Caution must be used in any support software for testing flight-critical systems. The support software must be tested sufficiently so that no errors appear that would mask errors in the critical software system under test.

REFERENCES

Anderson, D.C.; and Frank, D.A.: System Design and Analysis for Multimode Control Laws. AFTI/F-16 Development and Integration Program, DFCS Phase Final Technical Report, vol. 3, pt. 1, AFWAL-TR-84-3008, Air Force Flight Dynamics Laboratory, 1984.

Brooks, F.P., Jr.: The Mythical Man-Month, Essays on Software Engineering. Addison-Wesley Publishing Company, 1975.

Ford, D.C.; Hewitt, D.C.; Allen, M.G.; Kirk, J.E.; and Leddy, A.M.: AFTI/F-16 Development and Integration Program. AFWAL-TR-84-3008-6, Flight Testing. Part 1: Summary Flutter and Aeroservoelasticity, and Flying Qualities, 1984.

Gordoa, M.J., Jr.; Johnston, A.M.; Griswold, M.R.; and Hornsby, B.L.: Integrated Systems Simulation and On-Airplane Testing. AFTI/F-16 Development and Integration Program, DFCS Phase Final Technical Report, vol. 4, AFWAL-TR-84-3008, Air Force Flight Dynamics Laboratory, 1984.

Ishmael, S.D.; Regenie, V.A.; and Mackall, D.A.: Design Implications From AFTI/F-16 Flight Test. NASA TM-86026, 1984.

Joyner, M.L.: AFTI/F-16 Digital Flight Control System Evaluation. Air Force Flight Test Center, Edwards Air Force Base, California, AFFTC-TR-83-48, 1983.

Mackall, D.A.: AFTI/F-16 Digital Flight Control System Experience, in NASA Aircraft Controls Research. Proceedings of a Workshop at NASA Langley Research Center, Hampton, Virginia, Oct. 25-27, 1983, Beasley, G.P., compiler, 1983.

Price, W.T.; Grandia, M.J.; Boulware, J.M.; Jones, J.O.; and Yousey, W.J.: Configuration Design. AFTI/F-16 Development and Integration Program, DFCS Phase Final Technical Report, vol. 2., AFWAL-TR-84-3008, Air Force Flight Dynamics Laboratory, 1984.

Szalai, K.J.; Jarvis, C.R.; Krier, G.E.; Megna, V.A.; Brock, L.D.; and O'Donnell, R.N.: Digital Fly-By-Wire Flight Control Validation Experience. NASA TM-72860, 1978.

U.S. Department of Defense: Flight Control Systems — Design, Installation and Test of Piloted Aircraft. Military Specification MIL-F-9490D, 1975.

U.S. Department of Defense: Flying Qualities of Piloted Airplanes. Military Specification MIL-F-8785C. Wright-Patterson Air Force Base, Ohio, 1980.

U.S. Department of Defense: Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs. Military Standard MIL-STD-483A, 1985a.

U.S. Department of Defense: Specification Practices. Military Standard MIL-STD-490A, 1985b.

Yousey, W.J.; Schindler, T.M.; Johnson, A.M.; and Toles, R.D.: System Design and Analysis for Redundancy Management. AFTI/F-16 Development and Integration Program, DFCS Phase Final Technical Report, vol. 3, pt. 2, AFWAL-TR-84-3008, Air Force Flight Dynamics Laboratory, 1984.

TABLE 1. — DECOUPLED CONTROL REQUIREMENTS

	Flight condition 1[a]	Flight condition 2[b]
Fuselage pointing control		
Pitch pointing, deg	±2.5	±2.0
Azimuth pointing, deg	±3.0	±3.0
Direct force control		
Lift force control, g	1.0	1.5
Side force control, g	0.5	0.8

[a]Flight condition 1: 1- and 4-g maneuvering load conditions at Mach 0.6 at 5,000 ft.

[b]Flight condition 2: 1- and 4-g maneuvering load conditions at Mach 0.9 at 20,000 ft.

TABLE 2. — RELIABILITY AND FAULT-TOLERANCE REQUIREMENTS

Reliability requirements	
DFCS failure rate resulting in loss of control	1 in 10^7 flight hr, excluding power actuators hydraulics and independent backup unit
DFCS abort rate	1 in 10^5 flight hr
Fail-operational requirements	
First failure	Fully operational
Second failure of similar device	At least safe flight (Operational State III, MIL-F-9490D; U.S. Department of Defense, 1975); probability of 0.95 of fully operational
Switching	
Mode switching	Hands-on positive switching required to return to normal mode
Air-to-air mode switching	Hands on
Transients	
Switching transients	Negligible
Failure transients	Magnitude and duration of DFCS transients shall not introduce unsafe transient vehicle responses
Cooling requirements	
Flight control computers	Capable of sustained reliable operation without reliance on forced air cooling

TABLE 3. — DFCS COMPONENTS, REDUNDANCY, AND FAIL-OPERATIONAL CAPABILITY

Function or component	Redundancy	Capability
1. Stability and command augmentation electronics	Triple	Two fail-operative with successful self-test
2. Integrated servoactuator	Dual hydraulic and triple electrical input	Fail-operative, fail-safe with computer interface
3. DFCS hydraulics	Dual	Fail-operative
4. Mode select DFCS status	Dual	Fail-operative, fail-safe
5. Trim		
(A) Switches	Quadruple	Two fail-operative
(B) Electronics	Triple	Two fail-operative with successful self-test
6. Air data sensors		
(A) Static and impact pressures	Triple	Fail-operative, fail-safe with standby gains
(B) Angle of attack	Triple	Fail-operative, fail-safe with reconfiguration
(C) Angle of sideslip	Triple	Fail-operative, fail-safe with reconfiguration
7. Central air data computer	Single	
8. Leading-edge flap		
(A) Maneuver computation	Triple	Two fail-operative with successful self-test
(B) Command servo	Dual	Fail-operative at half rate
(C) Flap drive	Single	Asymmetry detection and shutoff
9. Stick sensors	Triple outputs with fourth active standby	Two fail-operative with active standby
10. Pitch, roll, and yaw rate sensors	Triple	Fail-operative, fail-safe with reconfiguration
11. Accelerometers	Triple	Fail-operative, fail-safe with reconfiguration

TABLE 4. — VERIFICATION CROSS-REFERENCE INDEX USED IN SYSTEM SPECIFICATION

Section 3 requirement reference		Verification methods					
		NA	1	2	3	4	5
3.1	Item definition		X				
3.1.1	Interface diagram	X					
3.1.2	Interface definition	X					
3.1.2.1	System interface	X					
3.1.2.2	Digital-fly-by-wire system interface					X	
3.1.2.3	Pilot-vehicle interface					X	
3.1.3	Major components list		X				
3.1.4	Government furnished property list	X					
3.2	Characteristics						
3.2.1	Performance characteristics						
3.2.1.1	General	X					
3.2.1.2	Specific			X	X		X
3.2.1.2.1	Direct force control					X	X
3.2.1.2.2	Weapon line-pointing					X	X
3.2.1.3	Stability and flying qualities			X		X	X
3.2.1.3.1	Normal mode			X		X	X
3.2.1.3.2	Departure and spin recovery			X		X	
3.2.1.3.3	Limitations			X		X	
3.2.1.3.4	Task-tailored flight modes			X		X	
3.2.1.3.5	Gain and phase margins			X		X	
3.2.1.3.6	Decoupled operations			X		X	X
3.2.1.4	Control law mechanization			X		X	
3.2.1.4.1	Multimode control			X		X	
3.2.1.4.2	Reconfiguration			X		X	
3.2.1.5	Redundancy management			X		X	
3.2.2	Physical characteristics						
3.2.2.1	System functional character			X		X	
3.2.2.2	Flight control computer complex		X				
3.2.2.3	DFCS power supplies		X			X	
3.2.2.4	DFCS sensors			X		X	
3.2.2.5	Aircraft sideslip sensing		X				
3.2.2.6	Pilot controllers			X		X	
3.2.2.6.1	Controller characteristics				X	X	X

NA - not applicable; verification methods: 1 - inspection, 2 - analysis, 3 — demonstration, 4 - ground test, 5 - flight test.

TABLE 4. — CONTINUED

Section 3 requirement reference		Verification methods					
		NA	1	2	3	4	5
3.2.2.6.2	Primary controller		X			X	
3.2.2.6.3	Secondary controller		X			X	
3.2.2.7	Trim				X	X	
3.2.2.8	DFCS caution and warning annunciation				X		
3.2.2.9	System weight				X		
3.2.2.10	Controlled surface actuators			X		X	
3.2.2.11	Independent backup			X		X	
3.2.2.12	DFCS software		X				
3.2.3	Reliability						
3.2.3.1	Failure rate, loss of control			X			
3.2.3.2	DFCS abort rate			X			
3.2.4	Maintainability			X			
3.2.5	Environmental conditions					X	
3.2.6	Power requirements						
3.2.6.1	Electrical					X	X
3.2.6.2	Hydraulic					X	X
3.2.7	Transportability		X				
3.3	Design and construction		X				
3.3.1	Parts, materials, processes		X				
3.3.2	Electromagnetic interference and compatibility						
3.3.2.1	General			X		X	
3.3.2.2	Design requirements			X		X	
3.3.2.3	Installation and integration requirements					X	
3.3.2.4	Electrical bonding requirements					X	
3.3.2.5	Lightning protection					X	
3.3.3	Nameplates and product marking		X				
3.3.4	Workmanship		X				
3.3.5	Interchangeability		X				
3.3.6	Safety			X			
3.3.6.1	Safety, descending order of precedence			X			
3.3.6.2	Health and safety criteria			X			
3.3.6.2.1	Toxicity			X			
3.3.6.2.2	Electrical equipment hazard			X			

NA - not applicable; verification methods: 1 - inspection, 2 - analysis, 3 — demonstration, 4 - ground test, 5 - flight test.

TABLE 4. — CONCLUDED

Section 3 requirement reference		Verification methods					
		NA	1	2	3	4	5
3.3.6.2.3	Personnel hazard and safety			X			
3.3.7	Human performance and human engineering			X			
3.4	Documentation		X				
3.5	Logistics	X					
3.6	Precedence	X					
3.6.1	Precedence of documents	X					
3.6.2	Application of prior quality	X					

NA - not applicable; verification methods: 1 - inspection, 2 - analysis, 3 — demonstration, 4 - ground test, 5 - flight test.

TABLE 5. — ANALOG AND DISCRETE INPUTS AND OUTPUTS

Analog inputs	
Azimuth error	Left angle of attack
Azimuth error rate	Left canard position
Beta aft	Left flaperon position
Beta delta pressure	Left horizontal tail position
Beta fore	Left main landing gear tachometer
Data age	Normal accelerometer
Demodulated left canard position	Pitch rate gyro
Demodulated left flaperon position	Pitch rate gyro speed detect
Demodulated left horizontal tail position	Pitch stick command
Demodulated right canard position	Pitch stick fourth transducer
Demodulated right flaperon position	Redundancy management test input
Demodulated right horizontal tail position	Right angle of attack
Demodulated rudder position	Right canard position
Elevation error	Right flaperon position
Elevation error rate	Right horizontal tail position
Impact pressure (Q_c)	Right main landing gear tachometer
Indicated side-mounted angle of attack	Roll rate amplified
Lateral accelerometer	Roll rate gyro
Leading edge flap position	Roll rate gyro speed detect
Leading edge flap tachometer no. 1	Roll stick command
Leading edge flap tachometer no. 2	Roll stick fourth transducer
	Rudder pedal command
	Rudder pedal fourth transducer

TABLE 5. — CONTINUED

Analog inputs	
Rudder position	Throttle controller command
Spare dc input no. 1	Yaw rate gyro
Static pressure (P_S)	Yaw rate gyro speed detect
Analog outputs	
Angle-of-attack side mount, instrumentation	Left horizontal tail command, primary servo valves
Beta delta pressure, instrumentation	Left horizontal tail command, secondary servo valves
Demodulated pitch rate output	Right angle-of-attack output
Demodulated roll rate output	Right canard command, primary servo valves
Demodulated yaw rate output	Right canard command, secondary servo valves
FLCC temperature, instrumentation	Right flaperon command, primary servo valves
Leading edge flap command	Right flaperon command, secondary servo valves
Leading edge flap actuator command nos. 1 and 2	Right horizontal tail command, primary servo valves
Left angle-of-attack output	Right horizontal tail command, secondary servo valves
Left canard command, primary servo valves	Rudder command, primary servo valves
Left canard command, secondary servo valves	Rudder command, secondary servo valves
Left flaperon command, primary servo valves	
Left flaperon command, secondary servo valves	
Discrete inputs	
Aerial refuel door	Left flaperon ISA fail no. 2 (PS no. 2)
Alternate flap switch	Left horizontal tail ISA fail no. 1 (PS no. 1)
CADC good	Left horizontal tail ISA fail no. 2 (PS no. 2)
CCV engage switch	Main landing gear weight on wheels
Electrical reset	Manual pitch override engage switch
Gun firing logic	Nose landing gear door
Identity discrete no. 1, FLCC C	Nose landing gear weight on wheels
Indentity discrete no. 2, FLCC B	PLA (military power)
Identity parity, FLCC A	PLA (power idle)
IFFC analog data valid	Right canard ISA fail no. 1 (PS no. 1)
IFFC engage switch	Right canard ISA fail no. 2 (PS no. 2
Independent backup select switch	Right flaperon ISA fail no. 1 (PS no. 1)
Landing gear handle position	Right flaperon ISA fail no. 2 (PS no. 2)
Leading edge flap asymmetry brake	Right horizontal tail ISA fail no. 1 (PS no. 1)
LEF asymmetry brake power	
Left canard ISA fail no. 1 (PS no. 1)	
Left canard ISA fail no. 2 (PS no. 2)	
Left flaperon ISA fail no. 1 (PS no. 1)	

TABLE 5. — CONCLUDED

Discrete inputs	
Right horizontal tail ISA fail no. 2 (PS no. 2)	Trim left wing down stick
Rudder ISA fail no. 1 (PS no. 1)	Trim nose down panel
Rudder ISA fail No. 2 (PS no. 2)	Trim nose down stick
Servo reset	Trim nose left panel
Speed brake extend	Trim nose right panel
Speed brake retract	Trim nose up panel
Stick trim select switch	Trim nose up stick
Trim left wing down panel	Trim right wing down panel
	Trim right wing down stick

Discrete outputs	
Analog test	LHT centering (low)
CADC reset	LFLAP centering (high)
CADC test	LFLAP centering (low)
Dual DFCS faiL no. 1 (high)	Normal accelerometer torque
Dual DFCS fail no. 2 (high)	Pitch rate gyro torque
DFCS fail (high)	PSA test
DFCS ready	PSA test enable
IBU engage	Right flap centering (high)
Input discrete BIT test one	Right flap centering (low)
Input discrete BIT test zero	RHT centering (high)
ISA reset (high)	RHT centering (low)
ISA reset (low)	Right canard centering (high)
Lateral accelerometer torque	Right canard centering (low)
LEF lock no. 1	Roll rate gyro torque
LEF lock no. 2	Rudder centering (high)
Left canard centering (high)	Rudder centering (low)
Left canard centering (low)	Stall warning
LHT centering (high)	Yaw rate gyro torque

TABLE 6. — TYPES OF AVIONICS INFORMATION

Type	Description	From	Through	To
Pilot-DFCS parameters				
Mode requests	Allowed selection of different control modes	MPD	SMS	DFCS
BIT	Allowed pilot to initiate preflight BIT	MPD	SMS	DFCS
Memory	Ground only option to read computer memories for diagnostic purposes	MPD	SMS	DFCS
Fault display	Allowed pilot to obtain detailed information about failure lights	MPD	SMS	DFCS
DFCS mode	Indication of actual DFCS mode engaged	DFCS	SMS	MPD
Miscellaneous data	BIT, memory and fault data requested by the pilot	DFCS	SMS	MPD
Control law parameters				
Pitch and roll attitude	Inputs to a G-bias function that assisted the pilot during rolls	INU	--	DFCS
Aircraft velocity		INU	--	DFCS
Instrumentation parameters				
Parameter location	Identifies 64 DFCS parameters for output from the DFCS to the instrumentation system	FCC	--	DFCS
Instrumentation parameters		DFCS	--	Instrumentation system

TABLE 7. — SUMMARY OF DFCS DISPLAYS IN THE COCKPIT

Name	Description
Dedicated failure lights	
FCS fail	Indicates a failure involving one level of redundancy
Dual fail	A failure involving two levels of redundancy
IBU	The IBU is engaged
HUD indications	
CCV	Control configured vehicle — indicates decoupled control modes are active
G-limit	Indicates that the preselected normal acceleration limit is active
Multipurpose displays	
Base page	Allowed for control mode selection and access to the fault, data, test, preset, and authority pages
Fault page	Allowed display and reset of DFCS failures
Data page	The data page has the same functions as the base page with additional data displays
Test page	Provided ability to read DFCS memory and initiate BIT, ground operation only
Preset page	Allowed changing the default relationship of pilot controllers to control functions
Authority page	Allowed pilot to set a normal acceleration limit, for flight-test purposes

TABLE 8. — MNEMONICS FOR MPD[a]

Level	Type	Class
Recon	All	IBU
1st	Pitch	Actuator
2nd	Roll	Branch
Lock	Yaw	Output
A/B	LHT	Compute
A	RHT	Input
B	L FLP	"Blank"
Center	R FLP	
"Blank"	L CND	
	R CND	
	Rudder	
	LEF	
	Air data	
	Switch	
	"Blank"	

[a]Level, type, and class are defined in table 9.

TABLE 9. — DESCRIPTION OF FAULT MNEMONICS

Level	
1st	A 1st failure of a particular type and class has occurred
2nd	A 2nd like failure of a particular type and class has occurred
Recon	Control law reconfiguration (recon) has occurred; will only appear if a 2nd like failure of a particular type and class can't be isolated
Lock	The leading edge flaps are locked
A	The secondary hydraulic system has failed
B	The primary hydraulic system has failed
A/B	A 2nd like hydraulic failure has occurred
Center	The displayed control surface has been centered
Type	
All	All inputs or outputs of a particular class are affected
Pitch	Pitch axis inputs have failed
Roll	Roll axis inputs have failed
Yaw	Yaw axis inputs have failed
LHT	Left horizontal tail
RHT	Right horizontal tail
LFLP	Left trailing edge flap

TABLE 9. — CONCLUDED

RFLP	Right trailing edge flap
LCND	Left canard
RCND	Right canard
Rudder	Rudder
LEF	Leading edge flap
Air data	Impact or static sensor has failed
Switch	A cockpit or aircraft switch has failed
Class	
IBU	Independent backup flight control system has failed
Actuator	An integrated servoactuator (ISA) has failed
Branch	All computer inputs and outputs in one flight control computer have failed
Output	There has been an output electronics failure in a flight control computer
Compute	There has been a computational failure in a flight control computer
Input	A sensor or controller has failed

TABLE 10. — NUMERIC CODES FOR FAULT DISPLAYS

Device identification number (DID)	Failure	Level	Type	Class
1	FLCC	1st, 2nd	All	Branch
2	D-A converter	1st, 2nd	All	Output
3	LHT total computed output	1st, 2nd	LHT	Compute
4	LHT coil wraparound	1st, 2nd	LHT	Output
5	RHT total computed output	1st, 2nd	RHT	Compute
6	RHT coil wraparound	1st, 2nd	RHT	Output
7	LFLP total computed output	1st, 2nd	LFLP	Compute
8	LFLP coil wraparound	1st, 2nd	LFLP	Output
9	RFLP total computed output	1st, 2nd	RFLP	Compute
10	RFLP coil wraparound	1st, 2nd	RFLP	Output
11	Rudder total computed output	1st, 2nd	Rudder	Output
12	Rudder coil wraparound	1st, 2nd	Rudder	Output
13	LCND total computed output	1st, 2nd	LCND	Compute
14	LCND coil wraparound	1st, 2nd	LCND	Output
15	RCND total computed output	1st, 2nd	RCND	Compute
16	RCND coil wraparound	1st, 2nd	LEF	Output
17	LEF total computed output	1st, 2nd	LEF	Compute
18	LEF hardware status	1st, 2nd	LEF	Output
19	A-D converter	1st, 2nd	All	Input

TABLE 10. — CONCLUDED

Device identification number (DID)	Failure	Level	Type	Class
20	800-Hz power supply (inverter)	1st, 2nd	All	Input
21	Pitch rate sensor	1st, 2nd recon	Pitch	Input
22	Spare			
23	Roll rate sensor	1st, 2nd recon	Roll	Input
24	Yaw rate sensor	1st, 2nd recon	Yaw	Input
25	Angle-of-attack sensor	1st, 2nd recon	Pitch	Input
26	Spare			
27	LEF pot wraparound	1st	Pitch	Input
28	Pitch stick	1st, 2nd	Pitch	Input
29	Roll stick	1st, 2nd	Roll	Input
30	Rudder pedal	1st, 2nd	Yaw	Input
31	Throttle twist	1st, 2nd recon	Pitch	Input
32	Normal acceleration sensor	1st, 2nd recon	Pitch	Input
33	Lateral directional acceleration sensor	1st, 2nd recon	Yaw	Input
34	Spare			
35	Static or impact sensor pressure	1st, 2nd recon	Air data	Input
36	Discrete IOC	1st, 2nd	Switch	Input
37	Discretes	1st, 2nd	Switch	Input
38	IBU pitch wraparound	1st, 2nd	Pitch	IBU
39	IBU lateral directional wraparound	1st, 2nd	Roll	IBU
40	LHT ISA pressure system	A, B, A-B, center	LHT	Actuator
41	RHT ISA pressure system	A, B, A-B, center	RHT	Actuator
42	LFLAP ISA pressure system	A, B, A-B, center	LFLP	Actuator
43	RFLAP ISA pressure system	A, B, A-B, center	RFLP	Actuator
44	Rudder ISA pressure system	A, B, A-B, center	Rudder	Actuator
45	LCND ISA pressure system	A, B, A-B, center	LCND	Actuator
46	RCND ISA pressure system	A, B, A-B, center	RCND	Actuator

TABLE 11. — DISCRETE INPUTS PROCESSED BY THE DISCRETE SELECTOR-MONITOR

Description	
Group 1: Requires 5 msec settling time	
Nose landing gear door	Pitch integrator inhibit
Landing gear handle	Right flap PS no. 1
Right horizontal tail PS no. 1	Right flap PS no. 2
Right horizontal tail PS no. 2	Speed break retract
Stick trim right wing down	Air refuel door open
CADC valid	LEF brake power
Left flap PS no. 1	PLA idle
Left flap PS no. 2	Stick trim left wing down
Rudder PS no. 1	Stick trim nose up
Rudder PS no. 2	Speed break extend
Right canard PS no. 1	Stick trim nose down
Gun firing	Weight on nose landing gear
CCV engage	PLA at military power
Right canard PS no. 2	LARAP engage request
IFFC engage	LARAP disengage request
IBU select	Left horizontal tail PS no. 2
Alternate flap switch	Left canard PS no. 1
Left horizontal tail PS no. 1	Left canard PS no. 2
LEF assymetry brake	IFFC good
Group 2: Requires 80 msec settling time	
Electrical reset	Panel trim nose down
Servo reset	Panel trim nose left
Panel trim nose up	Panel trim nose right
Manual pitch override engage	Stick trim disconnect
Panel trim left wing down	Weight on main landing gear
Panel trim right wing down	

TABLE 12. — ANALOG INPUTS PROCESSED BY THE SELECTOR-MONITOR

Input parameter	Redundancy level	Special requirements (see notes)	Selection algorithm usage	First fail persistence[a]	Second fail persistence[a]	Trip level (% FS)[b]
Angle of attack	3	2, 4	MSL[c]	7	7	15
Normal acceleration	3	2	GCA[d]	7	7	15
Pitch rate	3	2	GCA	7	7	15
Angle of side slip (beta)	1	5	NA[e]	NA	NA	NA
Roll rate	3	2	GCA	7	7	15
Yaw rate	3	2	GCA	7	7	15
Lateral acceleration	3	2	GCA	7	7	15
Static pressure	3	3, 7	GCA	7	7	15
Impact pressure	3	3, 8	GCA	7	7	10
IBU pitch wraparound	3	1, 3	NA	7	7	15
IBU lateral wraparound	3	1, 3	NA	7	7	15
Rudder pedals	4	1, 3, 6	GCAF[f]	7	NA	15
Roll stick	4	1, 3, 6	GCAF	7	NA	15
Throttle twist	3	3	GCAF	7	7	15
Pitch stick	4	1, 3, 6	GCAF	7	NA	15

Note 1 - no selection; input is monitored only.
Note 2 - selected and monitored immediately before control law execution, once per frame.
Note 3 - selected and monitored no more than one subframe before control law execution, once per frame.
Note 4 - must be corrected for flow effects prior to selection and monitoring.
Note 5 - must be corrected for flow effects prior to use.
Note 6 - second action of two-part selection and monitoring process.
Note 7 - limit selected value to $\geq$ 241.25 lb/ft^2.
Note 8 - limit selected value to positive value.

[a]Persistance - number of consecutive 15.6 msec frames required to fail an input.
[b]% FS - percentage of full-scale.
[c]MSL - median select logic.
[d]GCA - good channel average.
[e]NA - not applicable.
[f]GCAF - good channel average plus monitoring for a fourth input.

TABLE 13. — HIERARCHICAL STRUCTURE OF SYSTEM USED BY FAILURE MANAGER

1	FLCC (flight control computer)
1.1	Digital-to-analog (D-A) converter
1.1.1	Left horizontal tail coil wraparound
1.1.2	Right horizontal tail coil wraparound
1.1.3	Left flap coil wraparound
1.1.4	Right flap coil wraparound
1.1.5	Rudder coil wraparound
1.1.6	Left canard coil wraparound
1.1.7	Right canard coil wraparound
1.1.8	Leading edge flap (LEF) total computed output (TCO)
1.1.8.1	LEF hardware
1.2	Left horizontal tail TCO
1.3	Right horizontal tail TCO
1.4	Left flap TCO
1.5	Right flap TCO
1.6	Rudder TCO
1.7	Left canard TCO
1.8	Right canard TCO
2	Analog-to-digital (A-D) converter
2.1	Inverter (800-Hz power supply)
2.1.1	Pitch rate sensor
2.1.2	Roll rate sensor (normal or amplified)
2.1.3	Yaw rate sensor
2.1.4	Angle-of-attack sensor
2.1.5	Beta (yaw) sensor (at present not monitored)
2.1.6	LEF pot wraparound
2.1.7	Pitch stick
2.1.8	Roll stick
2.1.9	Rudder pedal
2.1.10	Throttle twist
2.2	Normal acceleration sensor
2.3	Lateral directional acceleration sensor
2.4	Impact pressure (Q_c) sensor or static pressure (P_s) sensor
3	Discrete IOC
3.1	Individual discretes (switches)
	IBU pitch wraparound
	IBU lateral directional wraparound
	Left horizontal tail ISA
	Right horizontal tail ISA
	Left flap ISA
	Right flap ISA
	Rudder ISA
	Left canard ISA
	Right canard ISA

TABLE 14. — SOFTWARE MECHANIZATION FOR GOOD-CHANNEL-AVERAGE SELECTOR MONITOR, NO PRIOR FAILURES

Values	Failure conditions									
ABS (L-S) > E	N	N	Y	Y	Y	N	N	Y	Y	Y
ABS (S-R) > E	N	Y	N	N	N	Y	Y	Y	Y	Y
ABS (L-R) > E	-	N	N	Y	Y	Y	Y	N	N	Y
LPC = first fail limit-1	-	-	-	N	Y	-	-	-	-	-
SPC = first fail limit-1	-	-	-	-	-	-	-	N	Y	-
RPC = first fail limit-1	-	-	-	-	-	N	Y	-	-	-
Required action	1	2	2	3	4	5	6	7	8	9

Actions:

```
1.  Decrement LPC if LPC > 0
              SPC if SPC > 0
              RPC if RPC > 0
    Select (L + S + R)/3

2.  Select (L + S + R)/3

3.  Decrement SPC if SPC > 0
              RPC if RPC > 0
    Increment LPC and TPC
    Select (S + R)/2

4.  Decrement SPC if SPC > 0
              RPC if RPC > 0
    Increment LPC and TPC
    Invoke failure manager
    Select based on DST MS

5.  Decrement LPC if LPC > 0
              SPC if SPC > 0
    Increment RPC and TPC
    Select (S + L)/2

6.  Decrement LPC if LPC > 0
              SPC if SPC > 0
    Increment RPC and TPC
    Invoke failure manager
    Select based on DST MS

7.  Decrement LPC if LPC > 0
              RPC if RPC > 0
    Increment SPC and TPC
    Select (L + R)/2

8.  Decrement LPC if LPC > 0
              RPC if RPC > 0
    Increment SPC and TPC
    Invoke failure manager
    Select based on DST MS

9.  Increment TPC
    Select (L + S + R)/3
```

Notes:

In all cases the selected output is 0 if the reconfiguration flag (RECF) is set.

L	left	XPC	persistance counters, number of iterations a failure has been present, where X is L, S, R, or T
S	self	E	fault detection level
R	right	ABS	absolute value
T	total	DST	device status table
		MS	monitor state

TABLE 15. — VERIFICATION TEST PLAN SUMMARY

Section			
1	Purpose		
2	Applicable documents		
	2.1	Government documents	
	2.2	Nongovernment documents	
3	Test concepts		
	3.1	Definition of terms	
	3.2	Description of program under test	
		3.2.1	System operational characteristics
		3.2.2	DFCS OFP functions
	3.3	Test philosophy	
4	Qualificaiton requirements and criteria		
	4.1	Stand-alone testing	
		4.1.1	AMUX data interface test
		4.1.2	Gain scheduler test
		4.1.3	Control law frequency response test
		4.1.4	Control mode selection and transition response test
		4.1.5	Analog input S/M operation test
		4.1.6	Discrete input S/M operation
		4.1.7	ISA monitor operation test
		4.1.8	LEF monitor operation test
		4.1.9	Long power outage test
		4.1.10	Short power outage test
		4.1.11	Memory and duty cycle reserve test
		4.1.12	Built-in test
		4.1.13	Flyable hardware retest
5	Test implementation		
	5.1	Location and schedule	
	5.2	Limitations and general comments	
	5.3	Preparation of input	
	5.4	Conduct of tests	
	5.5	Analysis of results	
	5.6	Summary of equipment	
	5.7	Special test software	
	5.8	Summary of personnel requirements	
6	Control and reporting procedures		
	6.1	Configuration control and documentation maintenance	
	6.2	Test failure analysis, repair and retesting	
7	Requirements cross-reference		

TABLE 16. — REVERIFICATION ACCEPTANCE TESTS

Test	
1	Verify AMUX input-output interface -Static SASB control law end-to-end calculations at Mach 0.9, sea level -AMUX input-output words
2	Verify end-to-end control law frequency response -Discrete sinusoidal inputs (1, 5, and 12 Hz) for all control modes -Pitch and yaw feedback sensor inputs
3	Verify multiple single fail-fault annunciation -Verify single fail device status entries for all monitoring planes in standard normal mode (SNRM) for analog inputs, discrete inputs, actuator inputs, ISA and LEF outputs
4	Verify first fail frequency response -Verifies SNRM first fail performance is same as no-fail performance (repeat test 2 for SNRM)
5	Verify single FLCC long outage restart performance -Satisfactory restart performance -Proper reconstruction of device status table fault history from non-volitive memory using test 3 faults
6	Verify restarted FLCC frequency response -Verify FLCC performance unaffected by a long power outage using test 2 for SNRM
7	Verify multiple dual fail and graceful degradation to one FLCC -Dual fail device status entries and AMUX fault annunciation in all monitoring planes -Insert parity error in one FLCC
8	Verify single FLCC frequency response test -Verifies remaining FLCC from test 7 has approximately same performance as nominal triplex system using test 2 for SNRM

TABLE 17. — INTEGRATED SYSTEM TESTS

Built-in test
Mode selection and display test
Control law frequency response
Step response test
Flight scenario test
Analog input single failure tolerance
Analog dual-like failure tolerance
ISA and LEF monitor failure tolerance
Analog multiple unlike input failure test
Stress test
Power outage-restart test
Control law gain margin test

TABLE 18. — FLYING QUALITIES TASKS AND CONTROL MODES

Task number	Subtest	Mode
1	Takeoff	SNRM
2	Air-to-air handling qualities	AAG
3	Air-to-air tracking	AAG
4	Decoupled air-to-air handling qualities	DAAG
5	Decoupled air-to-air tracking	DAAG
6	Air-to-surface tracking (bombs)	ASB
7	Decoupled air-to-surface tracking (bombs)	DASB
8	Air-to-surface tracking (guns)	ASG
9	Decoupled air-to-surface tracking (guns)	DASG
10	Power approach and landing	SNRM
		SNRM
11	Mode transients	All
12	α limiter	SNRM
		AAG
		DAAG

TABLE 19. — COMPONENTS CONSIDERED IN FAILURE MODES AND EFFECTS TESTING

1.	Engine failure	17.	ISA primary coil signal failure
2.	Emergency power unit (EPU) failure	18.	ISA secondary coil signal failure
3.	Main generator failure	19.	Leading-edge flap command failure
4.	System A hydraulic failure	20.	Input discrete section failure
5.	Any ISA servovalve SV1 or SV2 failure	21.	System input discrete failure
6.	System B hydraulic failure	22.	Wraparound input discrete failure
7.	Any ISA servovalve SV3 failure	23.	Output discrete section failure
8.	ISA solenoid valve failure	24.	ISA output discrete
9.	FLCC power supply failure	25.	Failure annunciation output discrete failure
10.	Central processing unit (CPU) failure	26.	BIT output discrete failure
11.	Input-output controller (IOC) failure	27.	Data link transmitter failure
12.	Analog-to-digital (A-D) converter failure	28.	Data link receiver failure
13.	800 Hz power failure	29.	Avionics multiplex bus failure
14.	Sensors-controllers	30.	IBU failure
15.	External ±15 V dc power failure	31.	LEF command servo
16.	Digital-to-analog (D-A) converter failure	32.	Runaway trim
		33.	Avionics failures
		34.	Digital value of actuator commands

TABLE 20. — FAILURE MODES AND THEIR EFFECTS

Component	Failure mode	Failure effect
Note: The first eight components were not considered to be DFCS components.		
9. FLCC power supply	Turn-off power	Branch failure including loss of one set of inputs, one CPU, and one set of outputs plus switching of one set of servovalve coils to the secondary coils
10. Central processing unit (CPU)	Halt CPU	Branch failure including loss of CPU and one set of outputs plus switching of one set of servovalve coils to the secondary coils

TABLE 20. — CONTINUED

Component	Failure mode	Failure effect
11. Input-output controller (IOC)	Halt IOC	Branch failure including loss of one set of inputs, one CPU, and one set of outputs plus switching of one set of servovalve coils to the seccondary coils
12. A-D converter	Hard-over A-D inputs	Ripple A-D and D-A failure trees
13. 20 V ac 800 Hz power	Turn-off power by pulling breaker to a branch inverter	Ripple the inverter resulting in loss of one set of 800 Hz inputs
14. Sensor and controller inputs	Turn-off any sensor or pilot controller input	Loss of a single input; other two inputs are monitored to obtain a valid signal
15. External ±15 V dc power	a. Ground +15 V dc from one FLCC b. Ground -15 V dc from one FLCC	a. Ripple A-D failure tree in one FLCC b. Ripple A-D failure tree in one FLCC
16. D-A converter	Hard-over D-A outputs	Ripple D-A resulting in loss of one set of outputs plus switching of one set of servovalve coils to the secondary coils
17. ISA primary coil	Open coil current wraparound	Coil current failure; switch to backup servoamplifier for driving secondary coil of one ISA
18. ISA secondary coil	Open backup coil current wraparound	Backup servo amplifier failure
19. LEF command	Hard-over command on FLCC C	LEF output electronic failure plus switching LEF drive to FLCC B
20. IOC input discrete section	Simultaneous failure of all system and wraparound discrete inputs to no change	Individual discretes failure in one FLCC each time a discrete input is changed by the pilot
21. System input discretes	Fail all system input discretes of one FLCC to undriven bus	Discrete failure resulting in loss of a partial set of system input discretes
22. Wraparound (W/A) input discretes	Incorporated in component no. 20	Not applicable

TABLE 20. — CONTINUED

Component	Failure mode	Failure effect
23. IOC output discrete section	Fail all BIT, ISA and failure annunciation output discretes of one FLCC to no change	Undetected first failure
24. ISA output discretes	Incorporated in component no. 23	Not applicable
25. Failure annunciation discretes	Incorporated in component no. 23	Not applicable
26. BIT output discretes	a. Landing gear handle discrete BIT inject	a. Lateral-directional IBU failure on landing
	b. Analog input BIT inject	b. Ripple A-D failure tree, plus switching to backup coils hardware in one FLCC
	c. IBU integrator BIT inject	c. Pitch IBU failure
27. Data link transmitter	Open both data link lines from FLCC B	Branch failure including loss of one set of inputs, one CPU, and one set of outputs plus switching of one set of servovalve coils to the secondary coils
28. Data link receiver	Fail one receiver of FLCC B to status good, data bad	Branch failure including loss of one set of inputs, one CPU, and one set of outputs plus switching of one set of servovalve coils to the secondary coils
29. AMUX	a. Status good, data bad on one bus	a. Loss of AMUX plus command to ASB mode
	b. Good data on both buses	b. Loss of AMUX and possibly one FLCC
	c. Bus contention	c. Indeterminate, possible FLCC loss
	d. Accepting any terminal address	d. Indeterminate, possible FLCC loss
30. IBU	Incorporated in component no. 26	Not applicable
31. LEF command servo	Kill power to one motor	Lock one motor drive

TABLE 20. — CONCLUDED

Component	Failure mode	Failure effect
32. Runaway trim	a. Nose-up b. Nose-down c. Right wing down d. Left wing down	Failure of stick trim switches
33. Avionic failures	a. FCC fail b. SMS fail	a. Force control of AMUX to SMS b. Loss of MPD
34. Digital value of actuator commands	a. Ramp on output of single surface output command, channel B b. Hard-over output of surface output commands by all computers	a. Simulated software error causes TCO miscompare b. Simulated generic software error requiring manual selection of the IBU

TABLE 21. — REVALIDATION TESTS

Test		
1	Verify DFCS and simulation interfaces pass preflight BIT	
2	Verify mode selection and other base page options	
	-MPD mode menu and CCV switches	-DGFT/MSOV/CCV switches
	-HUD mission phase mode and CCV switch	-Optimum flap-no scheduled flap
	-Drag modulation-drag conventional	-Flatturn decoupled-flatturn coupled
3	Verify fault annunciation of DFCS inputs	
	-Controller input failures	-Switch failures
	-Sensor input failures (first and second like)	
4	Verify preset decoupled options	
	-Preselected pedal, stick, throttle options for decoupled ASG modes	
	-Mode option changeability	
5	Verify pilot input discrete	
	-Landing gear up-down	-Panel trim
	-Nose gear door open-closed	-Stick trim
6	Validate take-off and landing performance	
	-Stores-clean (standard normal)	
	-Pitch rate reconfiguration (landing only)	
	-Standby gains (landing only)	
	-IBU (landing only)	
7	High-performance maneuvers	
	-Coupled maximum stick-rudder pedal-throttle twist commands (select modes)	
	-Maximum altitude-speed loops (all modes)	

TABLE 22. — IN-FLIGHT FAILURE INDICATIONS

Flight number	Number	Description	Cause	Correction
7	1	Indication of leading edge flap failure on touch and go	System integration	Vote software switches
15	2	Avionics forces rapid mode changes in flight control system	System integration (not resettable)	Avionics and flight control immunity increased
23	3	Indication of one of three DFCS branches failed; resettable by pilot	System integration	Vote software switches
23	4	Indication that one branch had failed to calculate horizontal tail commands correctly; resettable by pilot	Not repeatable	None
28	5	Indication that one branch had failed to calculate left and right canard commands; resettable by pilot	System integration	Vote software switch
36	6	Yaw departure results in failure indications for left and right canard actuators and air data; resettable by pilot. Investigation on air data failure mode identifies single failure that can cause loss of DFCS analog backup	System integration	Air data rate of change limited
44	7	Dual branch failure of DFCS, aircraft landed with single string control; failure not resettable	System integration	Vote software switch
54	8	Failure indication that one branch had failed to calculate its command to the flaperons; resettable by the pilot.	System integration	Vote software switch
66	9	Dual failure of an input discrete, traced to loose contacts in a connector	Hardware	Yes
82	10	Failure indication of left and right canard command in one branch; resettable by pilot	System integration	Discrete software switch

TABLE 22. — CONCLUDED

Flight number	Number	Description	Cause	Correction
85	11	Indication of one of three DFCS branches failed; resettable by pilot	System integration	Vote software switch
91	12	Indication of one of three DFCS branches failed; resettable by pilot; occurred during aircraft refueling	Unknown	None
Various flights from flight 72 to flight 100	13	Failure indications that an input discrete had failed; resettable by pilot; occurred five times	Unknown	None
Various flights	14	Failure indication that an input switch faded, occurred upon activation of a cockpit switch; resettable by pilot; occurred many times	Switch design	None
95	15	Failure indication of angle-of-attack sensor after flying through wake of another aircraft; resettable by pilot	Dissimiliar angle-of-attack sensors	None

TABLE 23. — BIT DETECTED FAILURES OF HARDWARE AND SOFTWARE

Number	Description
1	BIT detected a faulty relay used in switching commands to the leading edge flaps
2	BIT detected failures of semiconductor random access memory at approximately 40°F
3	BIT detected failure of a semiconductor discrete drive used in logic that detects second failures of the computers
4	BIT detected a failure of a hydraulic actuator
5	BIT detected a parity error and would not finish BIT test; the parity error indication resulted from a software error in the BIT test for parity errors

TABLE 24. — UNRESOLVED BIT FAILURE INDICATIONS

Number	Description
1	One of the three computers failed during BIT; suspect cause was loss of power to channel
2	BIT failed numerous times while testing various DFCS components; EMI was believed to be the cause

TABLE 25. — BIT FAILURES DUE TO SYSTEMS INTEGRATION

Number	Description
1	After tests to check battery power to the DFCS, BIT would detect failures of all input sensors
2	BIT software was unable to use back-up avionics bus in the event of a failure; resulted in pilot unable to monitor or operate BIT; system locked up
3	Accidental 2nd activation of BIT failed because of an unknown timing constraint for BIT operations
4	BIT detects leading edge flap lock and input failures because of improper procedures
5	BIT indicated failures because of noise induced from running actuator tests, numerous accounts
6	A procedure error causes BIT to be run with failures present; resulted in BIT locking up
7	BIT fails to detect a fault in the IBU after a modification to the IBU was made; hardware modification did not have corresponding change in BIT test

TABLE 26. — ANOMALIES WITH MEMORY MODE OPERATION

1.	One of three computers failed because of entering memory mode with control inputs; fault detection upon exit of memory mode causes interchannel differences
2.	Engine start in the memory mode caused complete DFCS failure and hard-over canards to impinge on nose wheel door

TABLE 27. — AERODYNAMIC FAILURE TRANSIENTS AS A FUNCTION OF MISSION MAXIMUM TRANSIENTS[a]

Configuration	Conditions, Mach (M), altitude (ft)	Normal acceleration, g	Angle of attack, deg	Angle of sideslip, deg	Roll angle, deg	Roll rate, deg/sec	Pitch rate, deg/sec
Air to air	M 0.9, 20,000 ft	0.55	0.55	0.77	3.3	28	7.7
Air to surface	M 0.6, 5,000 ft	0.55	0.55	0.77	4.4	33	7.7

[a]During first 200 msec after failure in which control surface(s) move away from recovery position.

TABLE 28. - PROPOSED CONTROL SURFACE FAILURE TRANSIENT SPECIFICATION

Altitude, ft	Mach number	Surface	Transients, deg
Sea level	0.95	Rudder	3
10,000	0.95	Rudder	5
40,000	0.95	Rudder	10
Sea level	0.95	Flaps	5
10,000	0.95	Flaps	7
40,000	0.95	Flaps	10

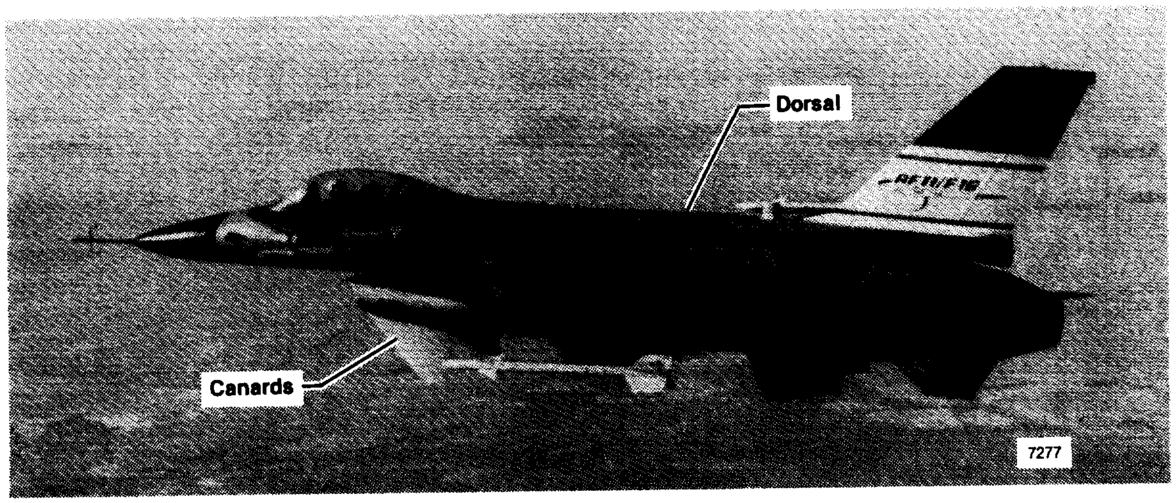


Figure 1. The AFTI F16 airplane.

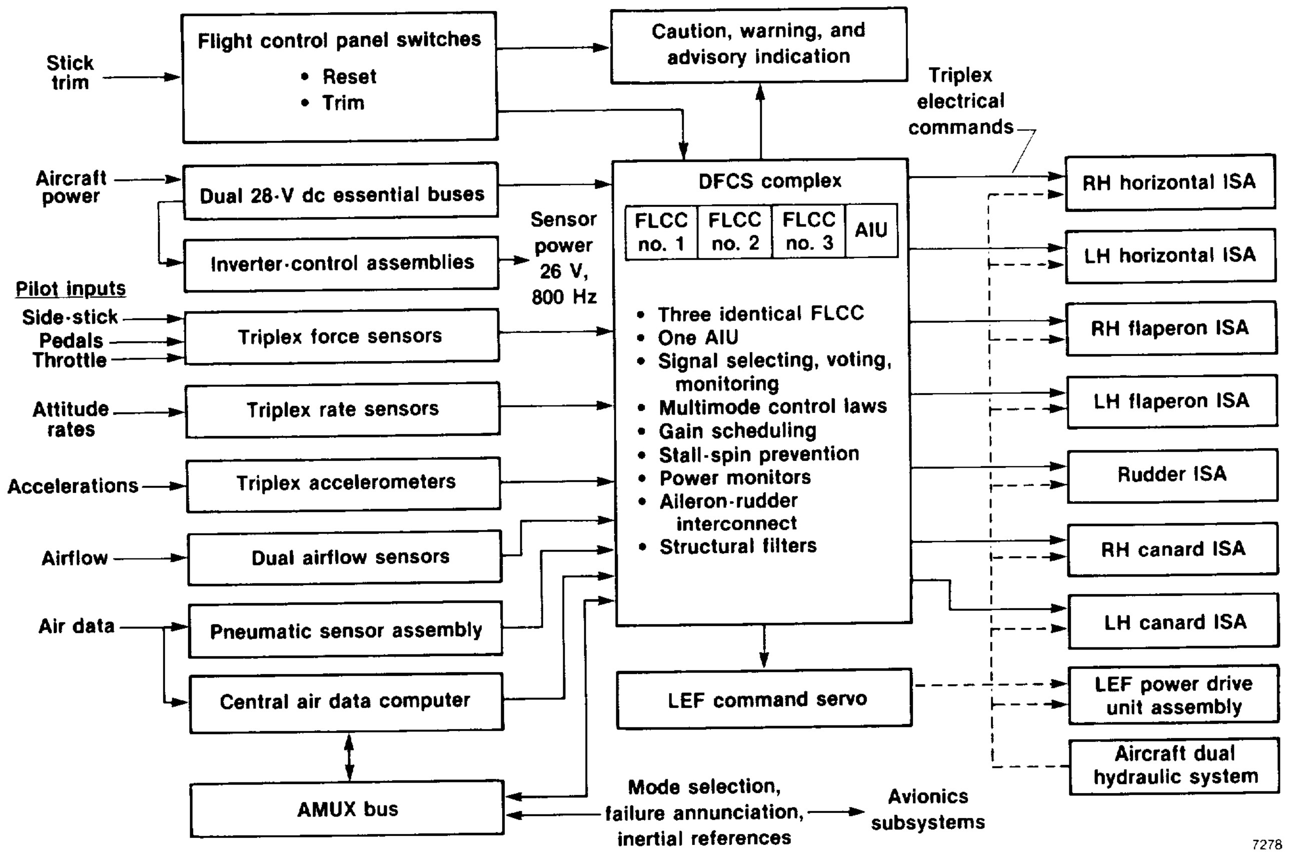


Figure 2. Digital flight control system architecture.

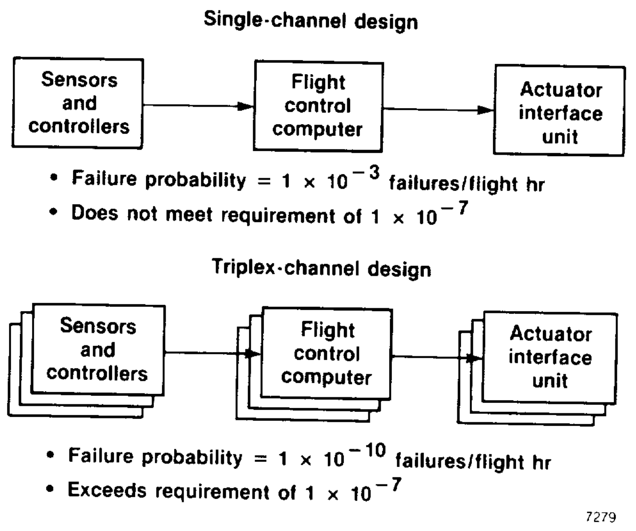


Figure 3. Reliability requirements force redundancy.

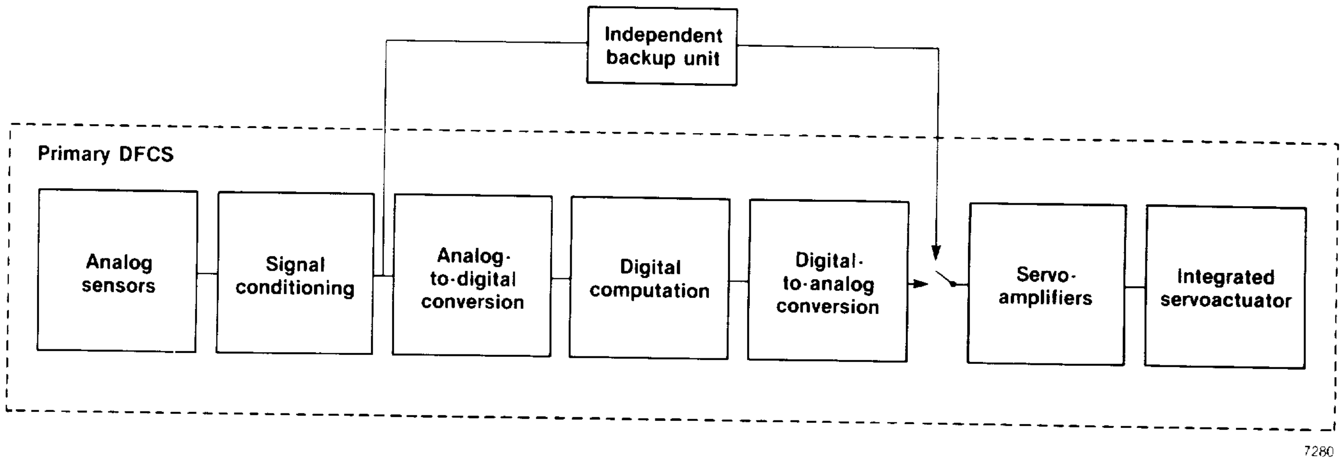


Figure 4. Independent back-up unit interface to the digital flight control system.

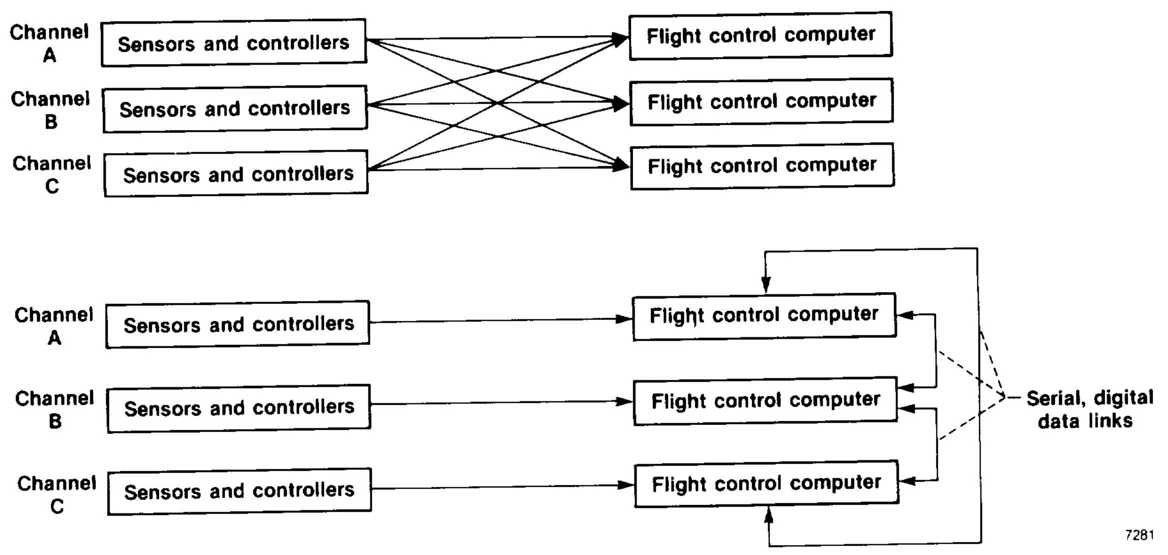


Figure 5. Architectural comparison of analog cross-strapping compared to digital cross-strapping.

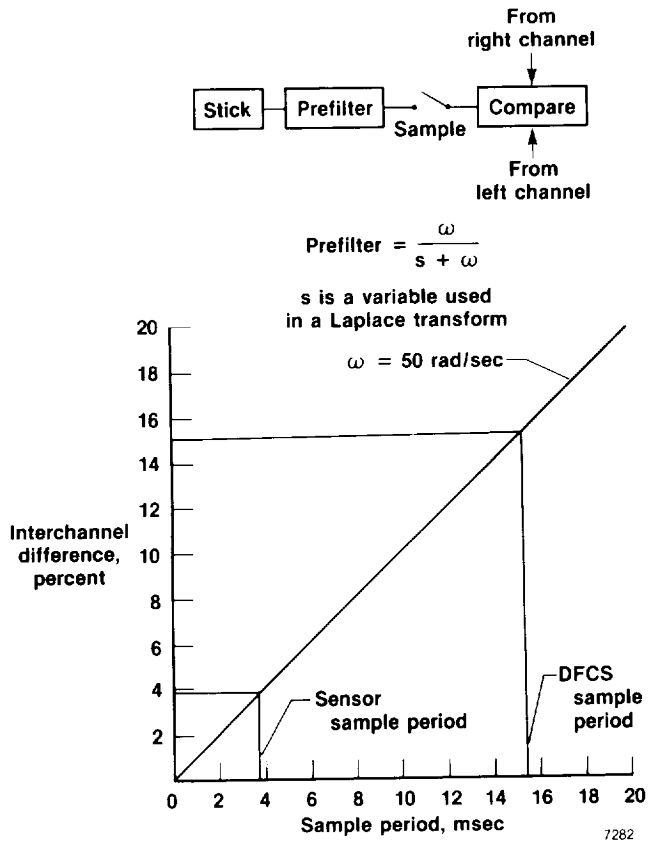
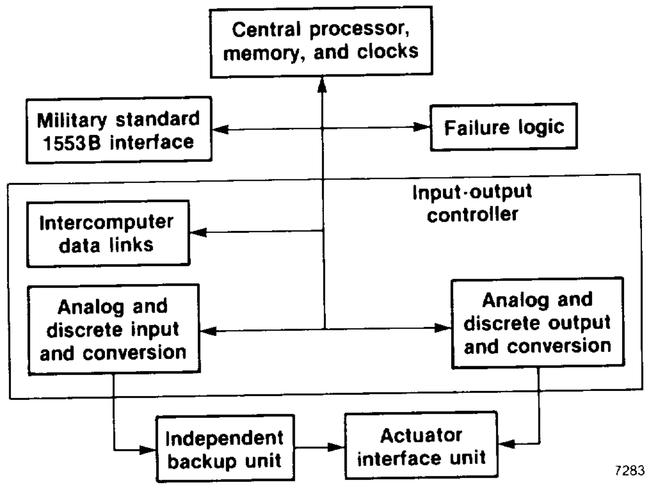


Figure 6. Interchannel difference for 1000 percent/sec ramp.



Central processor, memory, and clocks
Military standard 1553B interface
Failure logic
Input-output controller
Intercomputer data links
Analog and discrete input and conversion
Analog and discrete output and conversion
Independent backup unit
Actuator interface unit
7283

Figure 7. Flight control computer block diagram.

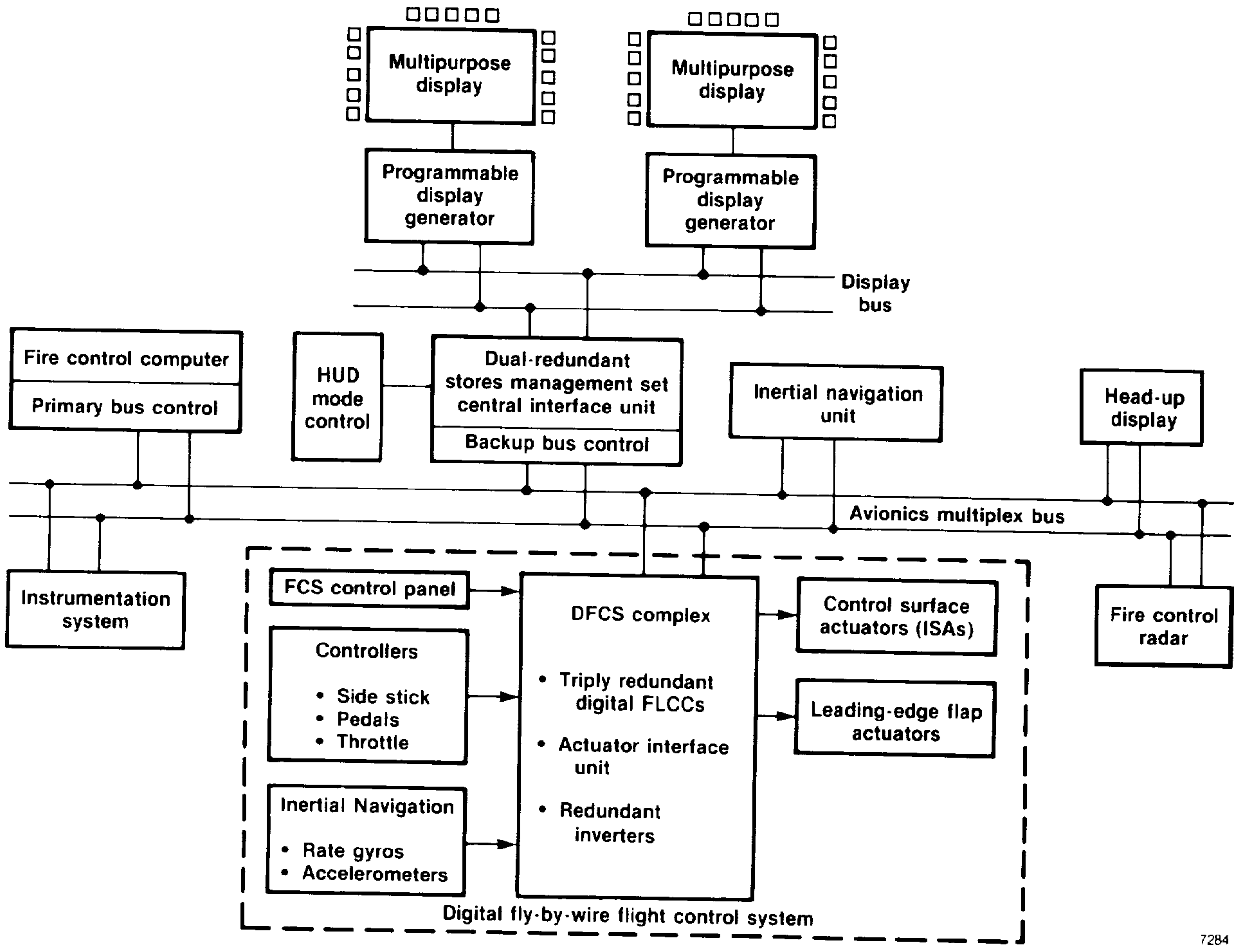


Figure 8. Digital flight control system and interface with avionics.

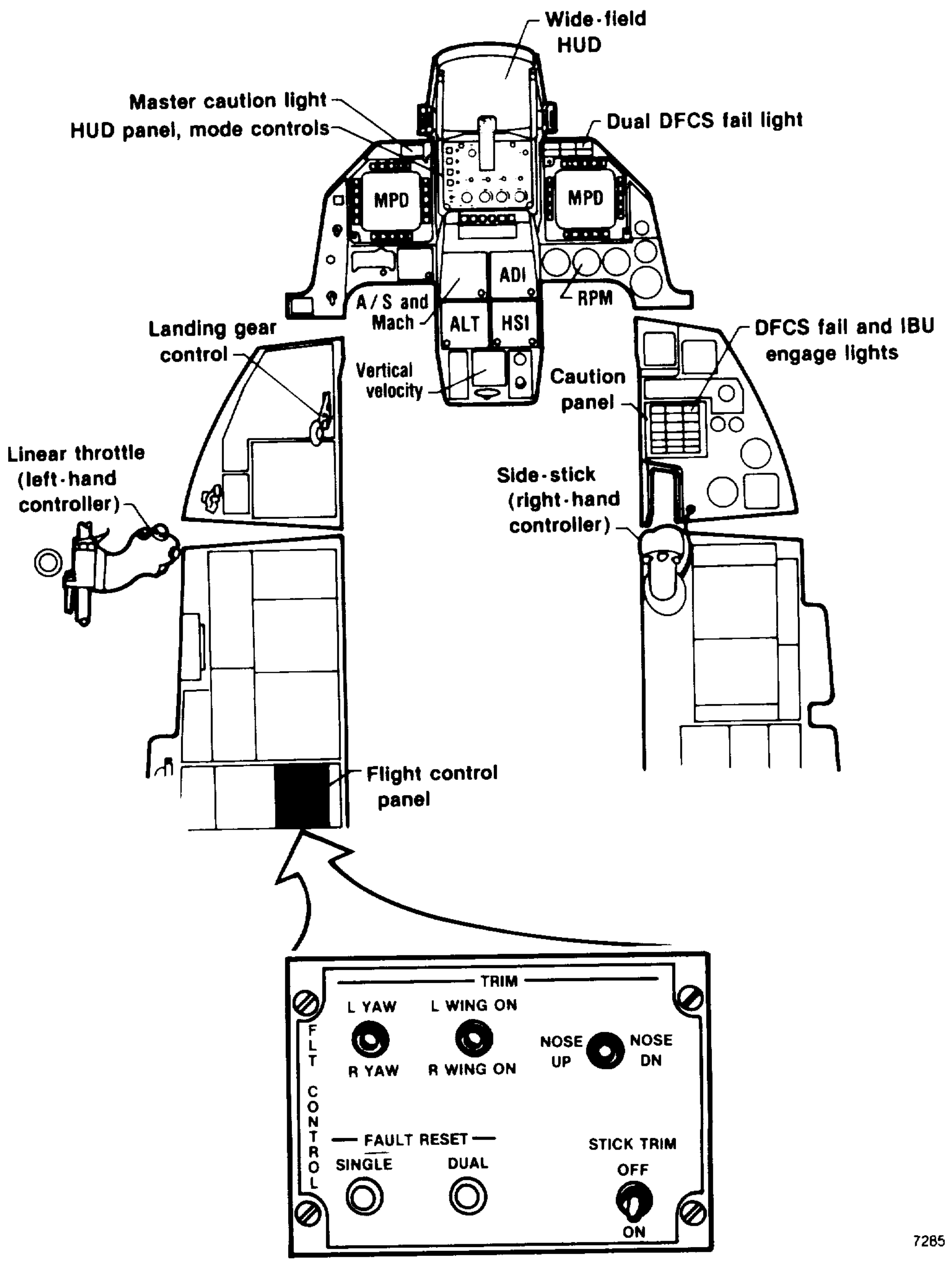
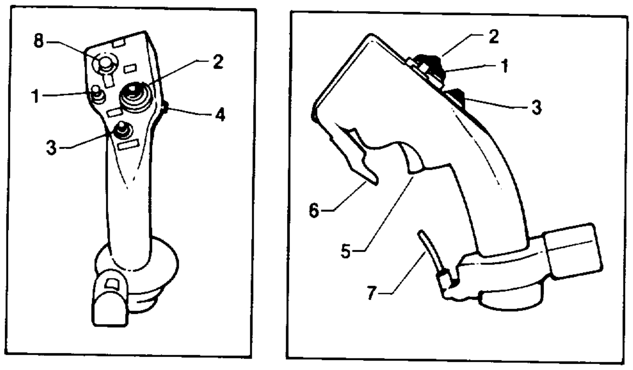


Figure 9. AFTI cockpit.



Pilot's side-stick controller

1. Weapons release	5. Record-laser-gun
2. Trim (pitch and roll)	6. CCV
3. Designate or return to search or helmet-mounted site	7. IFFC (DFCS stick limiting)
4. Nose wheel steering-air refueling disconnect-missile step	8. IBU

Functions common to F-16 and AFTI F-16

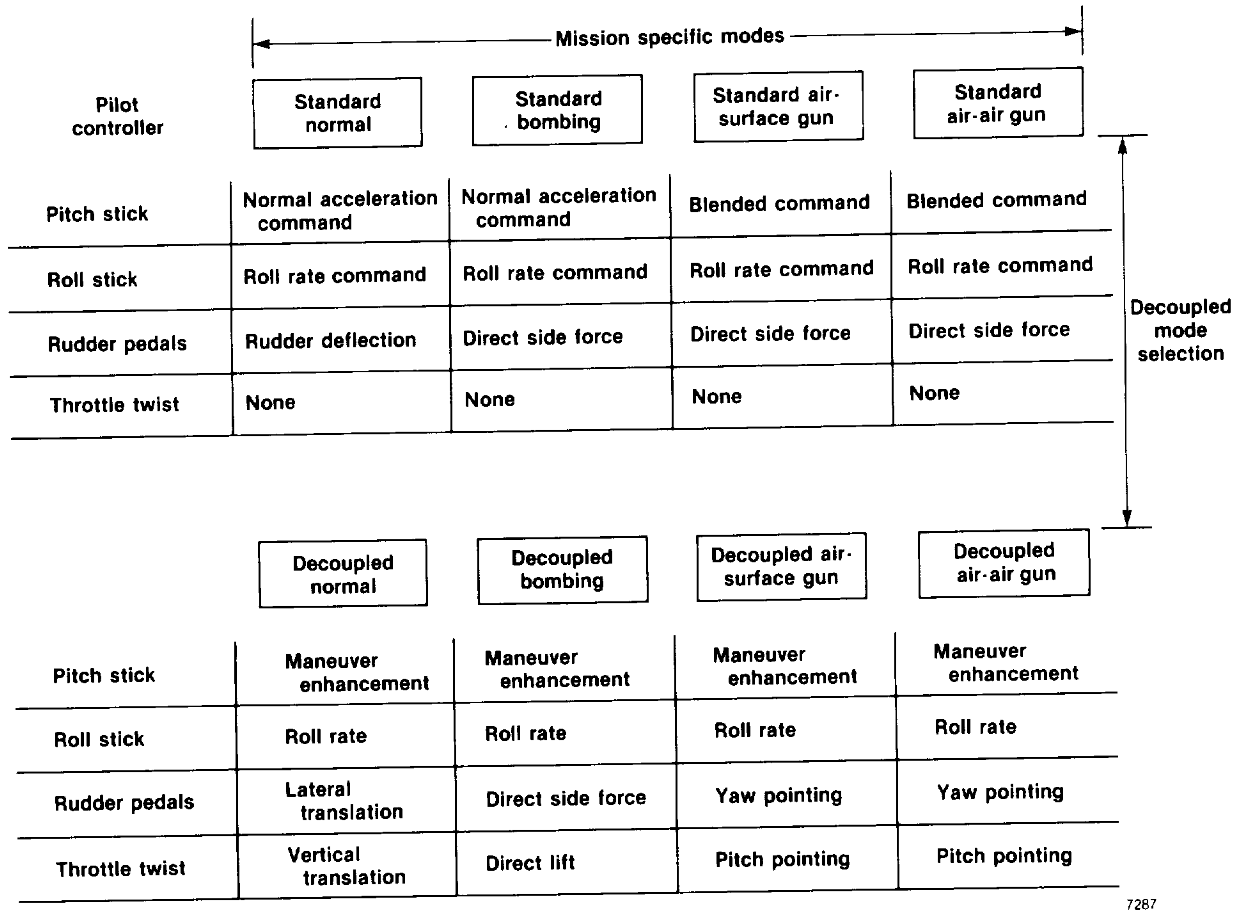
- Weapons release button
- Trim button (pitch and roll)
- Designate or return to search
- Nose wheel steering-air refuel disconnect-mean sea level step
- Camera-gun trigger

Additional functions peculiar to AFTI F-16

- CCV engage
- IBU engage
- IFFC engage

7286

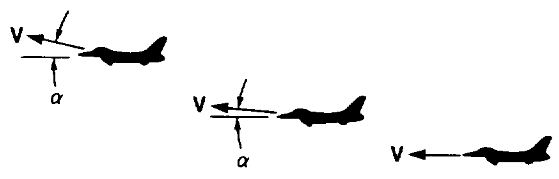
Figure 10. Right-hand controller.


Mission specific modes
Decoupled mode selection
7287

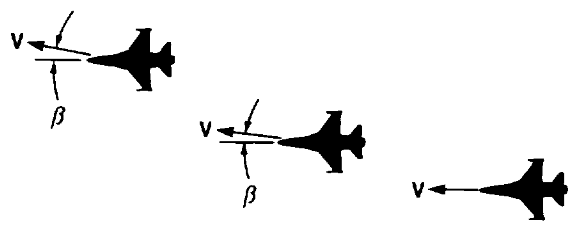
Pilot controller	Standard normal	Standard bombing	Standard air-surface gun	Standard air-air gun
Pitch stick	Normal acceleration command	Normal acceleration command	Blended command	Blended command
Roll stick	Roll rate command	Roll rate command	Roll rate command	Roll rate command
Rudder pedals	Rudder deflection	Direct side force	Direct side force	Direct side force
Throttle twist	None	None	None	None

	Decoupled normal	Decoupled bombing	Decoupled air-surface gun	Decoupled air-air gun
Pitch stick	Maneuver enhancement	Maneuver enhancement	Maneuver enhancement	Maneuver enhancement
Roll stick	Roll rate	Roll rate	Roll rate	Roll rate
Rudder pedals	Lateral translation	Direct side force	Yaw pointing	Yaw pointing
Throttle twist	Vertical translation	Direct lift	Pitch pointing	Pitch pointing

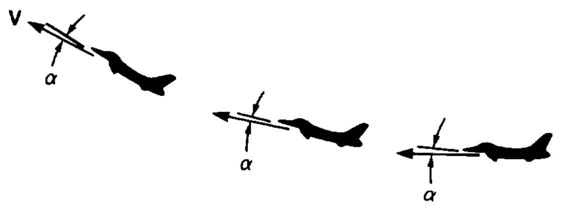
Figure 11. Control modes and controller commands.



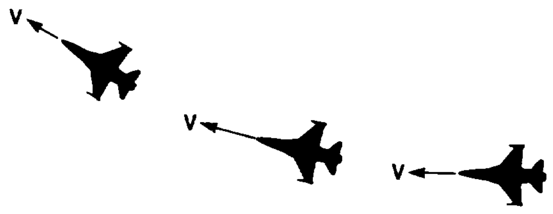
(a) Vertical translation: vertical velocity control at constant pitch attitude.



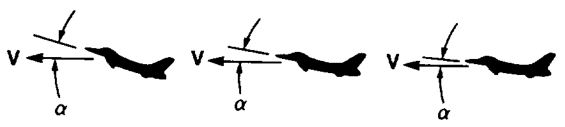
(d) Lateral translation: lateral velocity control at constant yaw attitude.



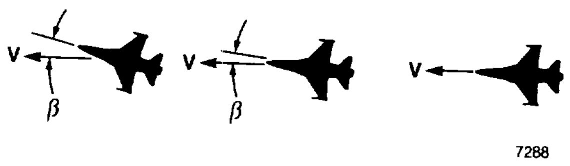
(b) Direct lift: vertical flightpath control at constant angle of attack.



(e) Direct sideforce: directional flightpath control at zero sideslip angle.



(c) Pitch pointing: pitch attitude control at constant flightpath angle.



(f) Yaw pointing: directional attitude control at constant flightpath angle.

Figure 12. Decoupled control descriptions.

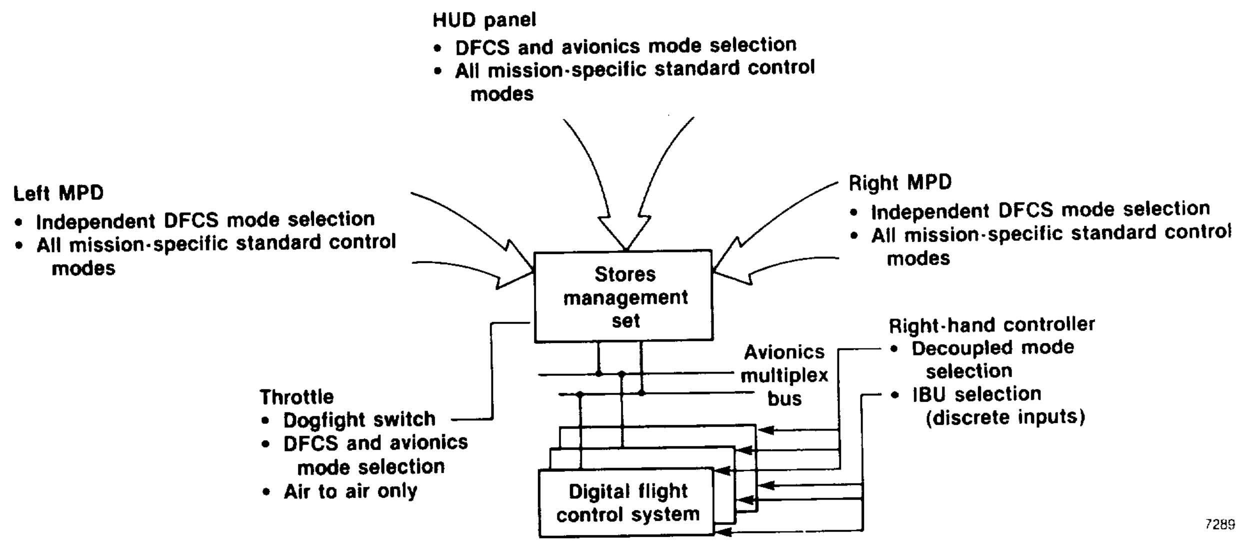
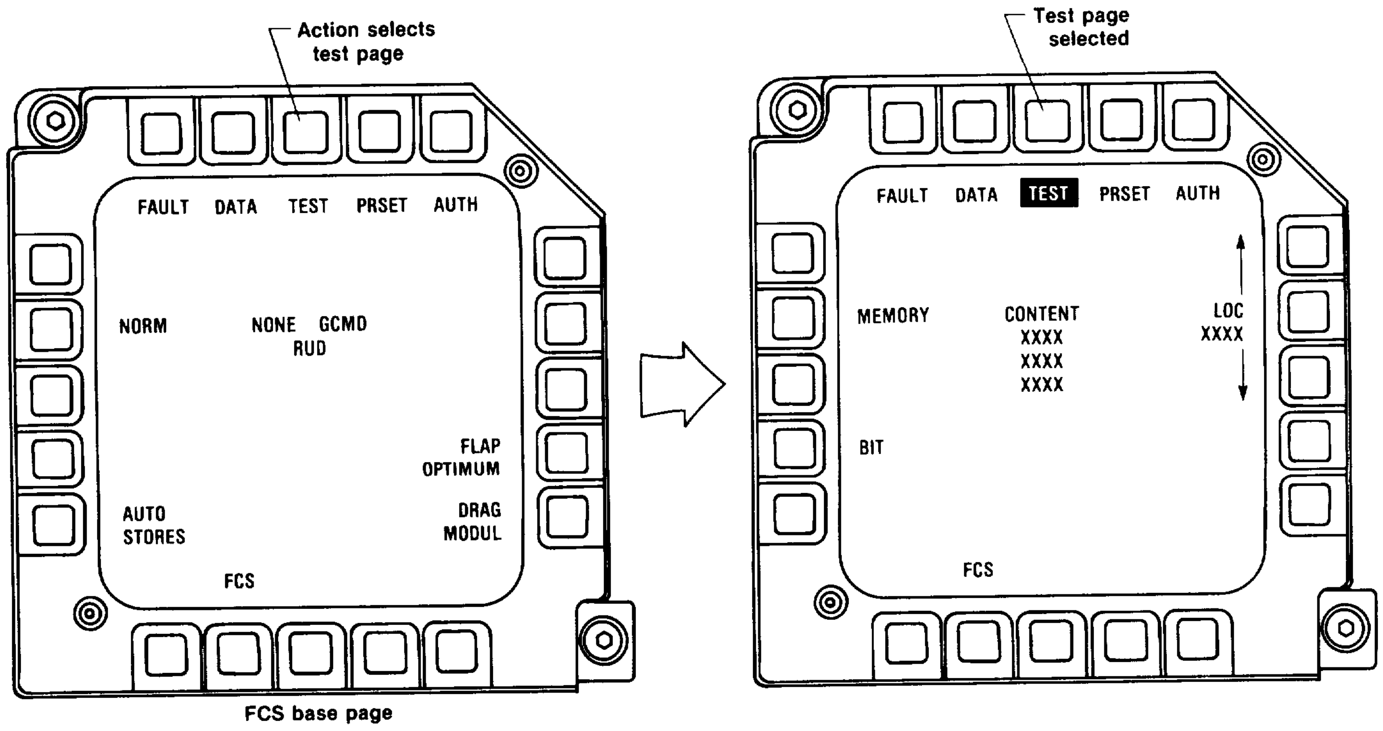
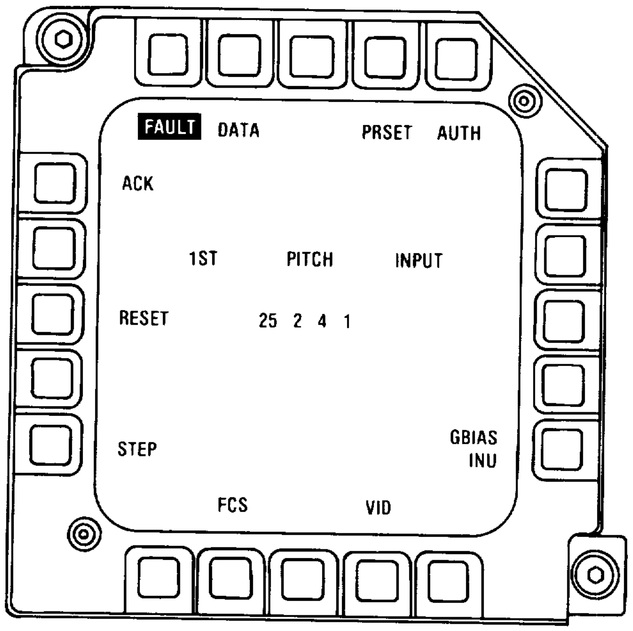

HUD panel
• DFCS and avionics mode selection
• All mission-specific standard control modes
Left MPD
• Independent DFCS mode selection
• All mission-specific standard control modes
Right MPD
• Independent DFCS mode selection
• All mission-specific standard control modes
Stores management set
Avionics multiplex bus
Right-hand controller
• Decoupled mode selection
• IBU selection (discrete inputs)
Throttle
• Dogfight switch
• DFCS and avionics mode selection
• Air to air only
Digital flight control system
7289

Figure 13. Flight control mode selection.



PRSET	preset	RUD	rudder
AUTH	authority	MODUL	modulation
NORM	normal	LOC	location
GCMD	G-command (normal acceleration)	FCS	flight control system

Figure 14. Multipurpose displays of digital flight control system base and test pages.



PRSET	preset
AUTH	authority
ACK	acknowledge
VID	video
INU	inertial navigation unit
GBIAS	G (normal acceleration) bias

7291

Figure 15. Multipurpose fault display.

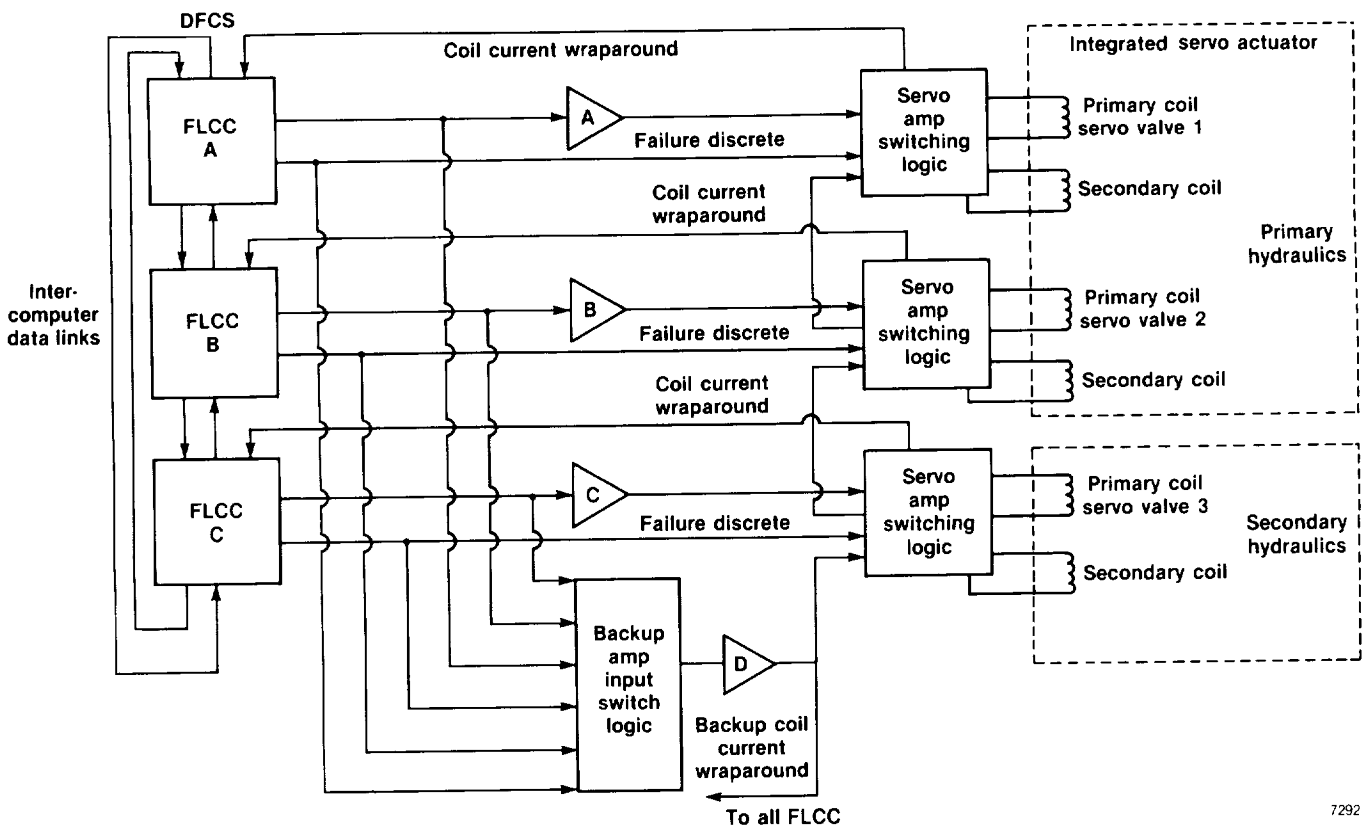
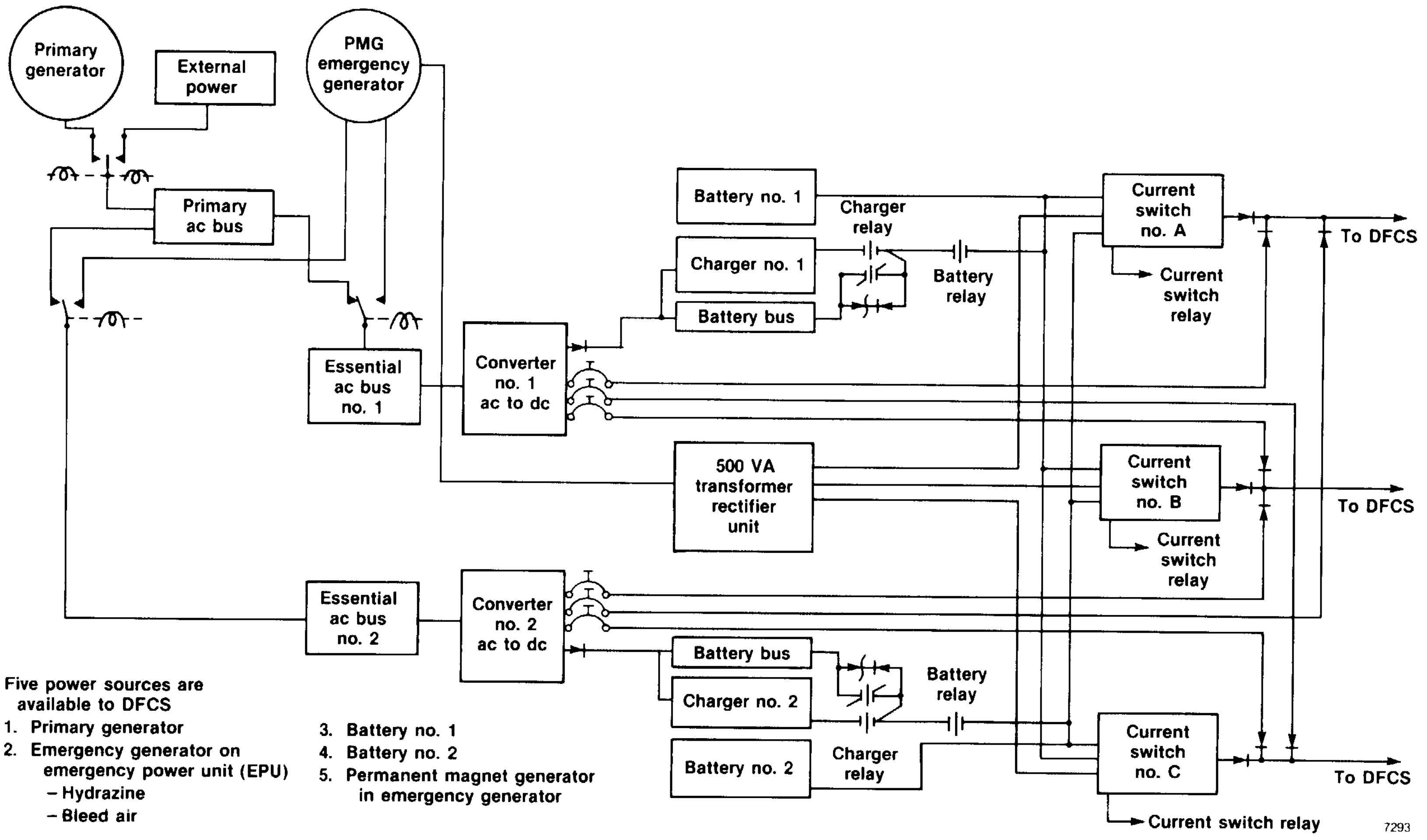

DFCS
Coil current wraparound
Integrated servo actuator
FLCC A
A
Failure discrete
Servo amp switching logic
Primary coil servo valve 1
Secondary coil
Coil current wraparound
Primary hydraulics
Inter-computer data links
FLCC B
B
Failure discrete
Servo amp switching logic
Primary coil servo valve 2
Secondary coil
Coil current wraparound
FLCC C
C
Failure discrete
Servo amp switching logic
Primary coil servo valve 3
Secondary coil
Secondary hydraulics
Backup amp input switch logic
D
Backup coil current wraparound
To all FLCC
7292

Figure 16. Actuator interface to digital flight control system.



Five power sources are available to DFCS

1. Primary generator
2. Emergency generator on emergency power unit (EPU)
   - Hydrazine
   - Bleed air
3. Battery no. 1
4. Battery no. 2
5. Permanent magnet generator in emergency generator

Figure 17. Electrical interface to digital flight control system preliminary design.

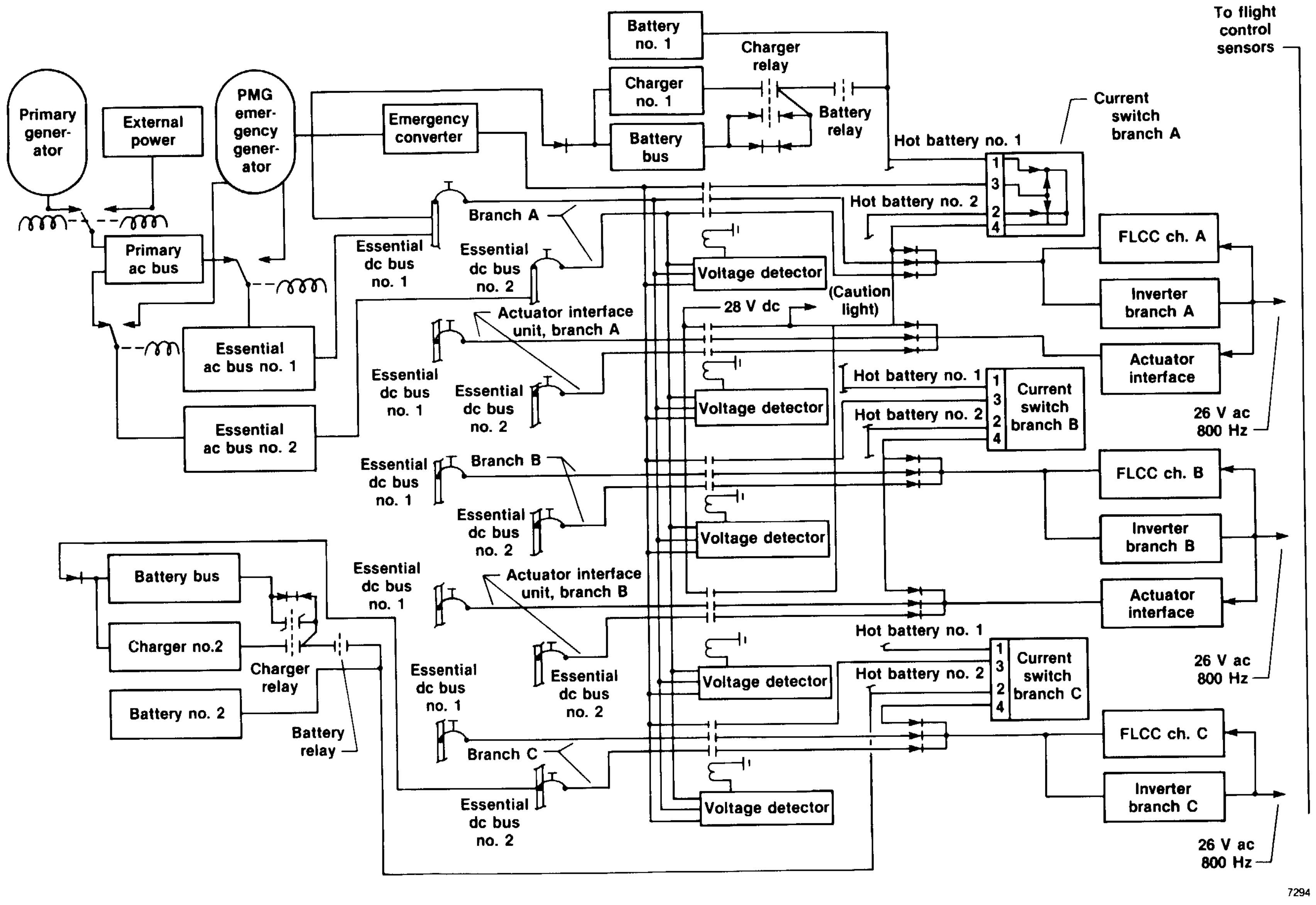


Figure 18. Final electrical power interface to digital flight control system.

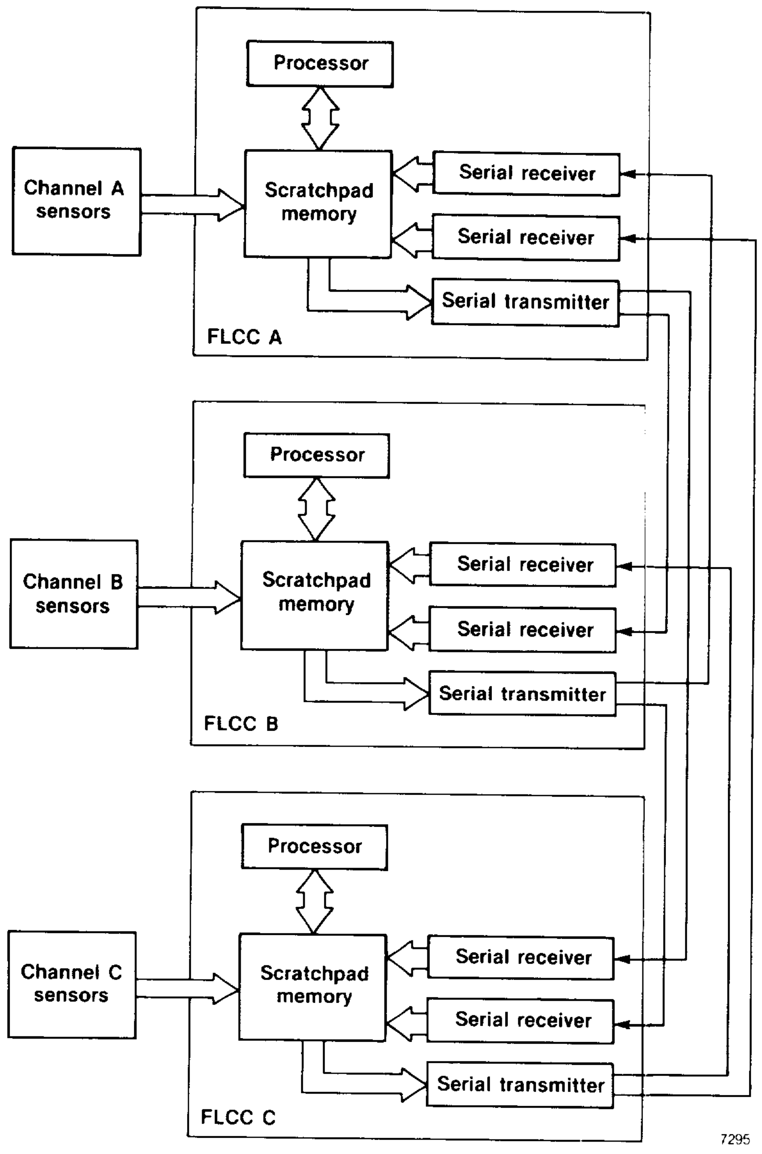


Figure 19. Cross-channel monitoring uses information sent on digital links.

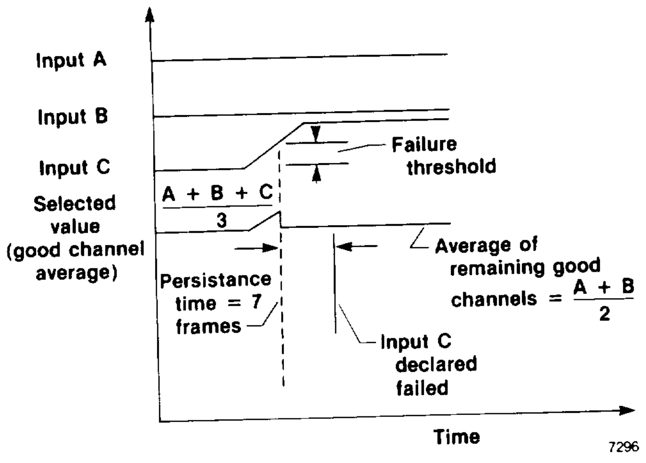
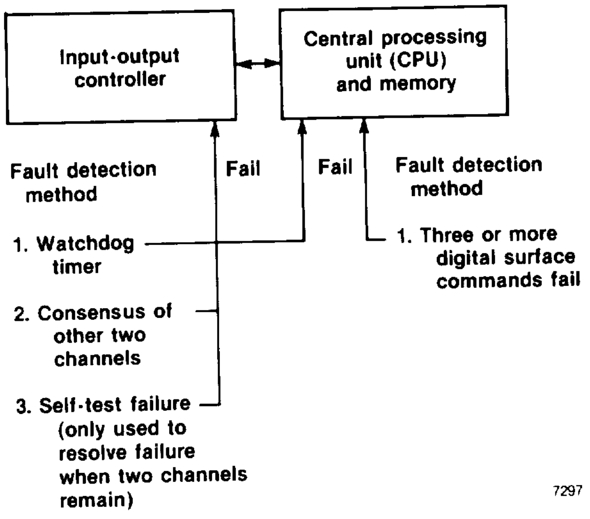
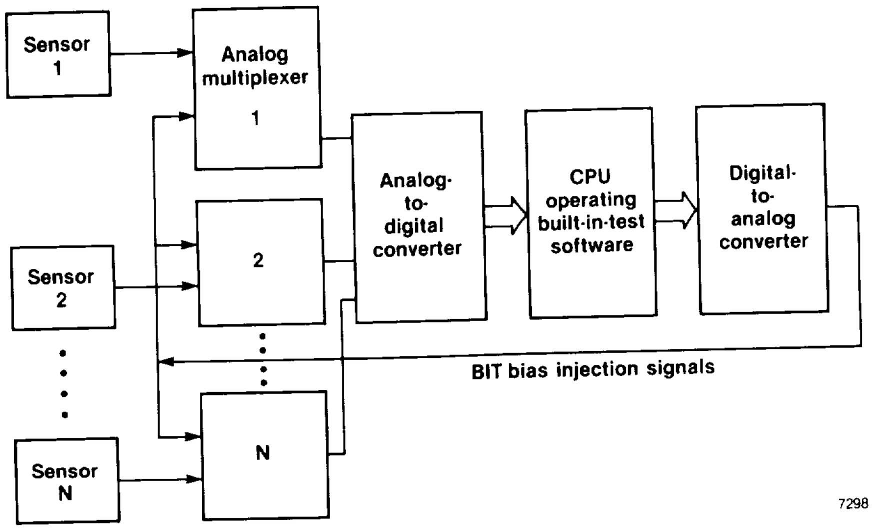


Figure 20. Overview of good-channel-average selection.



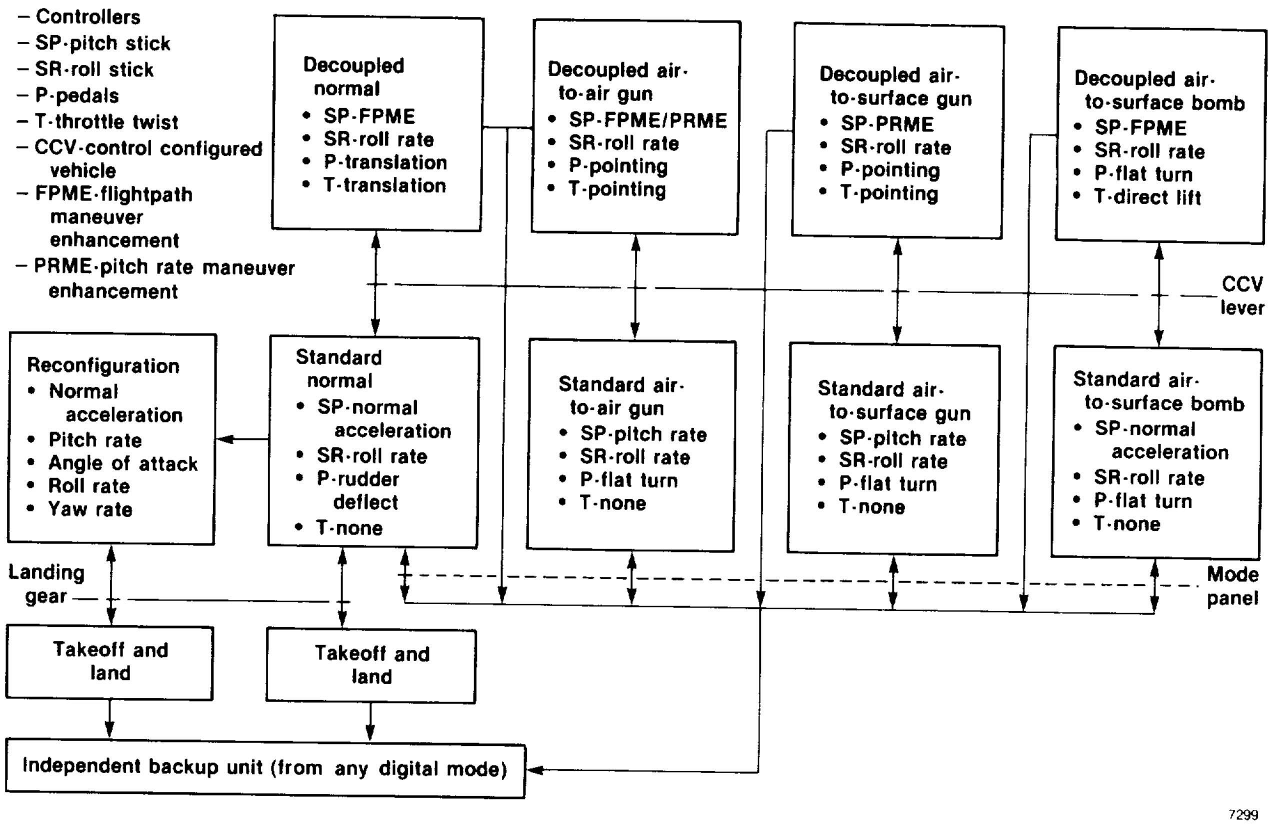
Input-output controller
Central processing unit (CPU) and memory
Fault detection method
1. Watchdog timer
2. Consensus of other two channels
3. Self-test failure (only used to resolve failure when two channels remain)
Fail
Fail
Fault detection method
1. Three or more digital surface commands fail
7297

Figure 21. Logic used to determine a channel failure.



Sensor 1
Sensor 2
Sensor N
Analog multiplexer 1
2
N
Analog-to-digital converter
CPU operating built-in-test software
Digital-to-analog converter
BIT bias injection signals
7298

Figure 22. Bias injection to resolve sensor and input circuitry faults.



– Controllers
– SP-pitch stick
– SR-roll stick
– P-pedals
– T-throttle twist
– CCV-control configured vehicle
– FPME-flightpath maneuver enhancement
– PRME-pitch rate maneuver enhancement
Decoupled normal • SP-FPME • SR-roll rate • P-translation • T-translation
Decoupled air-to-air gun • SP-FPME/PRME • SR-roll rate • P-pointing • T-pointing
Decoupled air-to-surface gun • SP-PRME • SR-roll rate • P-pointing • T-pointing
Decoupled air-to-surface bomb • SP-FPME • SR-roll rate • P-flat turn • T-direct lift
CCV lever
Reconfiguration • Normal acceleration • Pitch rate • Angle of attack • Roll rate • Yaw rate
Standard normal • SP-normal acceleration • SR-roll rate • P-rudder deflect • T-none
Standard air-to-air gun • SP-pitch rate • SR-roll rate • P-flat turn • T-none
Standard air-to-surface gun • SP-pitch rate • SR-roll rate • P-flat turn • T-none
Standard air-to-surface bomb • SP-normal acceleration • SR-roll rate • P-flat turn • T-none
Landing gear
Mode panel
Takeoff and land
Takeoff and land
Independent backup unit (from any digital mode)
7299

Figure 23. Standard and decoupled mission specific control structure.

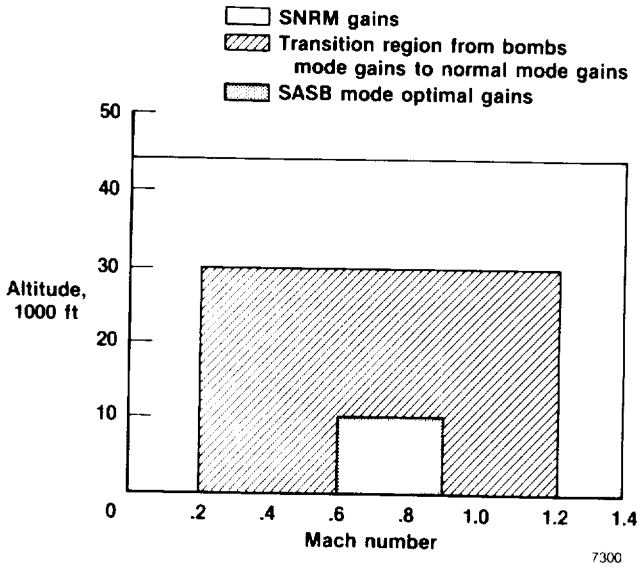


Figure 24. Bombing mode flight envelope and gain changes.

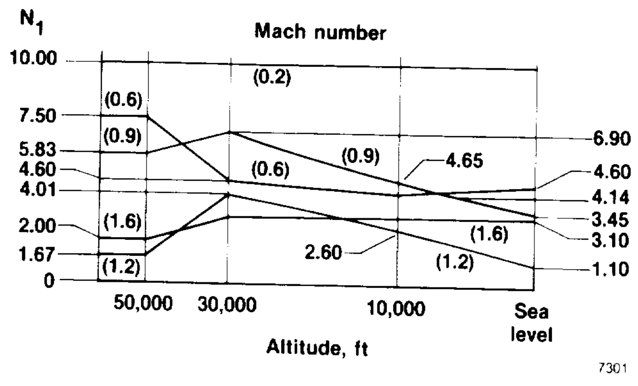


Figure 25. Longitudinal feed forward gain N_1 requiring double interpolation.

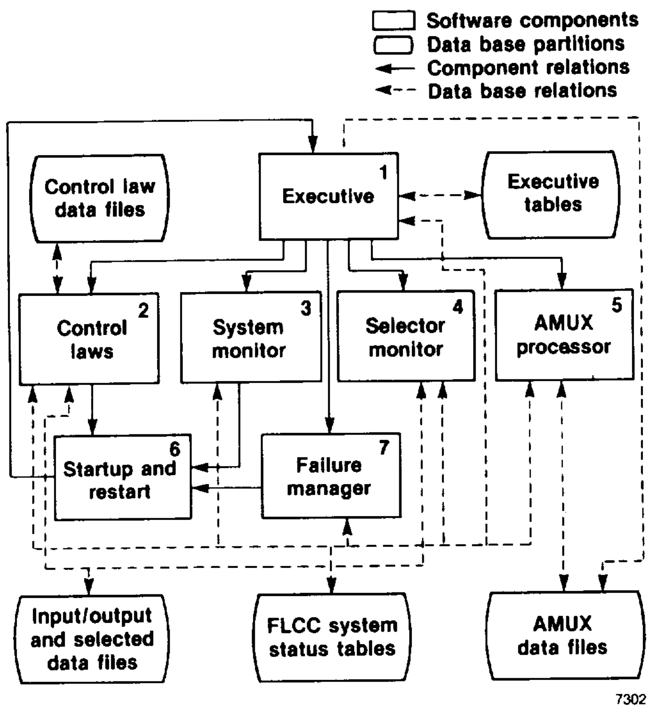
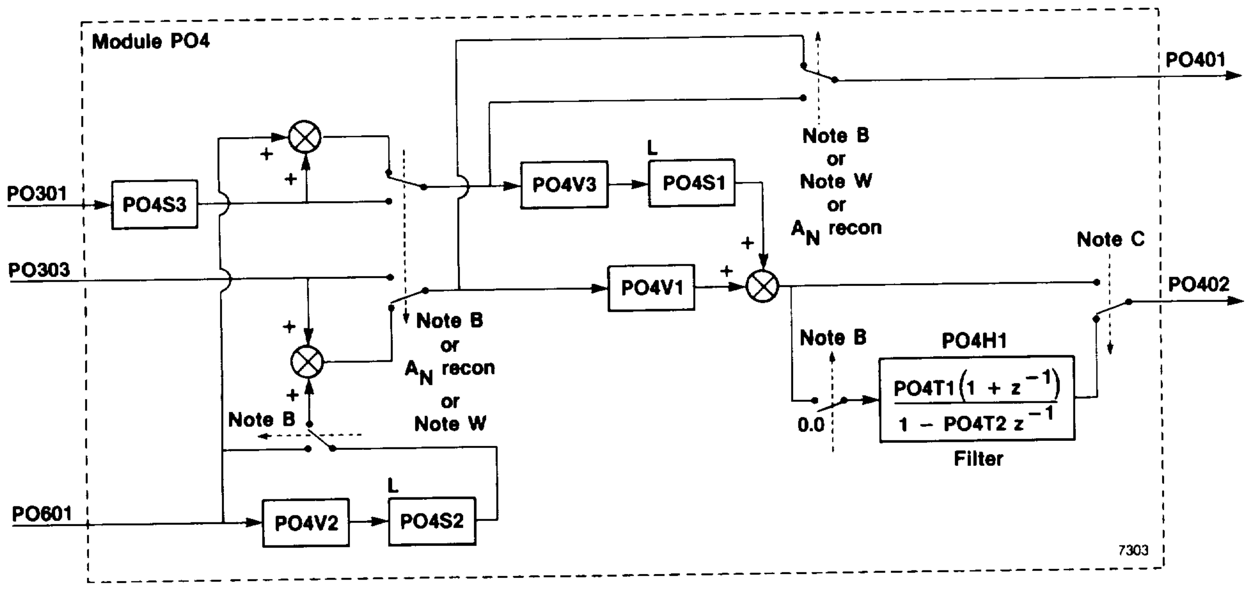


Figure 26. Digital flight control system software structure.



Module PO4
PO301
PO303
PO601
PO4S3
PO4V3
PO4S1
PO4V1
PO4V2
PO4S2
L
Note B or Note W or A_N recon
Note B or A_N recon or Note W
Note B
Note C
PO401
PO402
PO4H1
$\frac{PO4T1(1 + z^{-1})}{1 - PO4T2\, z^{-1}}$
0.0
Filter
7303

Figure 27. Software mechanization example.

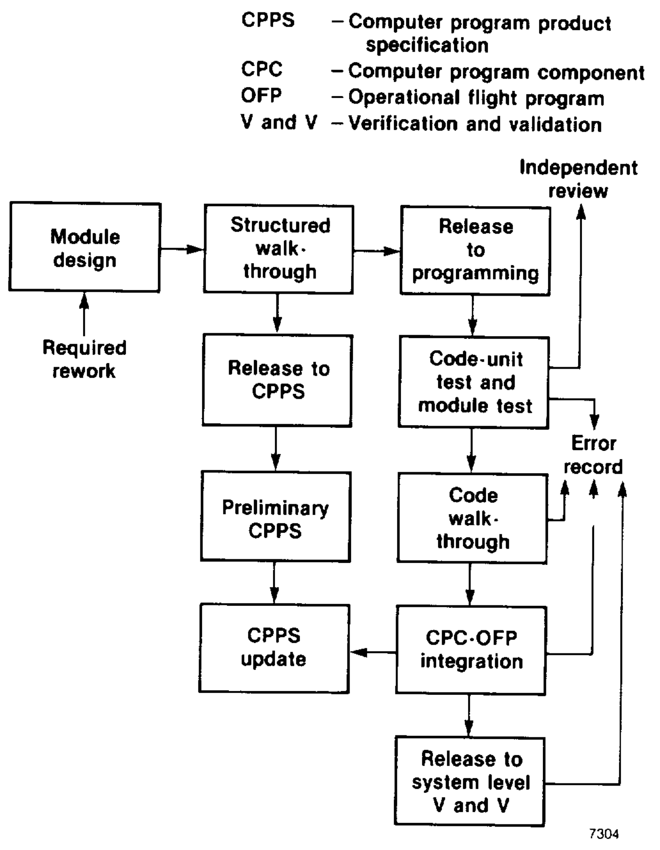

CPPS – Computer program product specification
CPC – Computer program component
OFP – Operational flight program
V and V – Verification and validation
Module design
Required rework
Structured walk-through
Release to programming
Independent review
Release to CPPS
Code-unit test and module test
Error record
Preliminary CPPS
Code walk-through
CPPS update
CPC-OFP integration
Release to system level V and V
7304

Figure 28. *Software design process.*

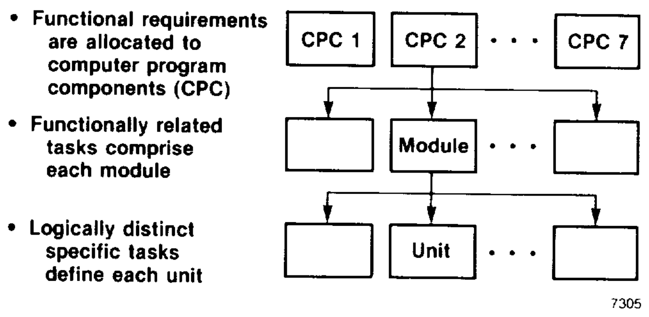

• Functional requirements are allocated to computer program components (CPC)
• Functionally related tasks comprise each module
• Logically distinct specific tasks define each unit
CPC 1
CPC 2
CPC 7
Module
Unit
7305

Figure 29. *Top-down software structure.*

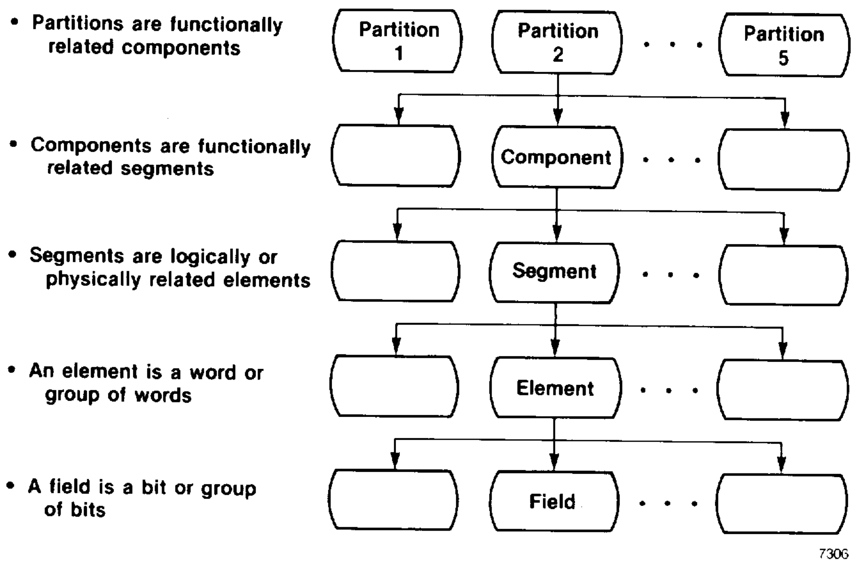

• Partitions are functionally related components
• Components are functionally related segments
• Segments are logically or physically related elements
• An element is a word or group of words
• A field is a bit or group of bits
Partition 1
Partition 2
Partition 5
Component
Segment
Element
Field
7306

Figure 30. *Software data base structure.*

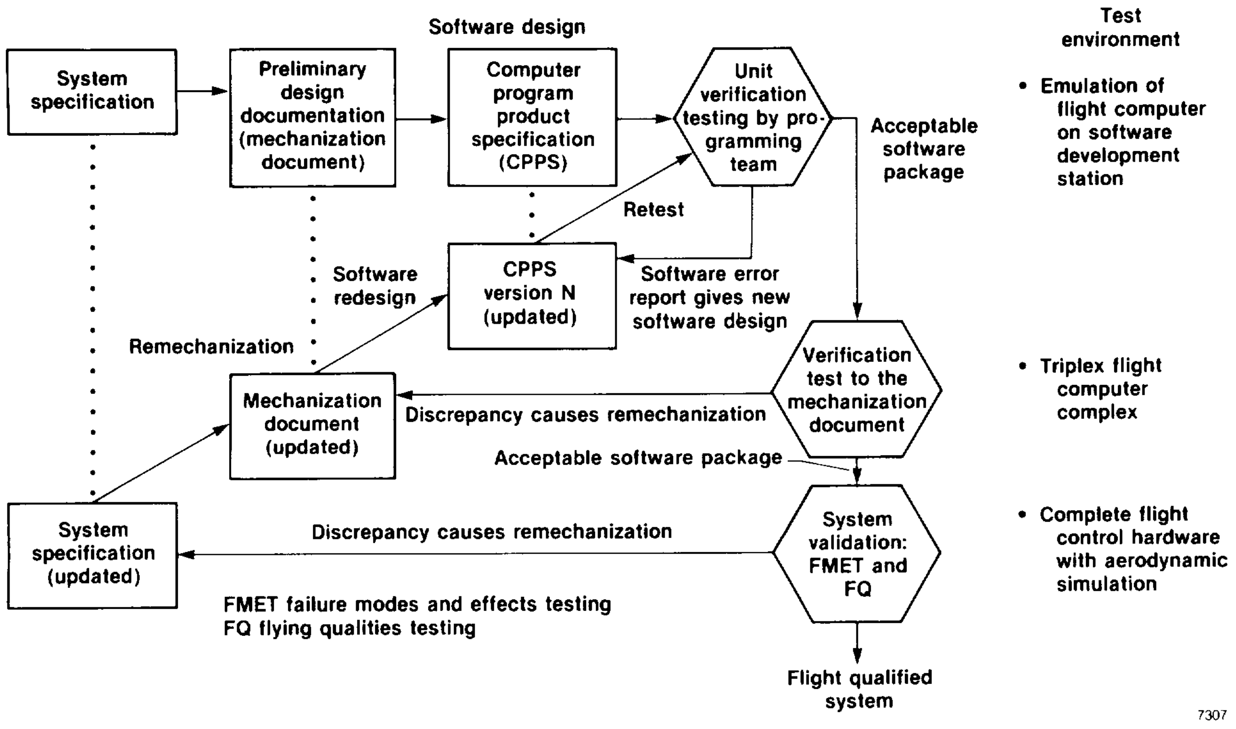

Software design
Test environment
System specification
Preliminary design documentation (mechanization document)
Computer program product specification (CPPS)
Unit verification testing by programming team
Acceptable software package
• Emulation of flight computer on software development station
Retest
CPPS version N (updated)
Software error report gives new software design
Software redesign
Remechanization
Verification test to the mechanization document
• Triplex flight computer complex
Mechanization document (updated)
Discrepancy causes remechanization
Acceptable software package
System specification (updated)
Discrepancy causes remechanization
System validation: FMET and FQ
• Complete flight control hardware with aerodynamic simulation
FMET failure modes and effects testing
FQ flying qualities testing
Flight qualified system
7307

Figure 31. System-software qualification and design iterations.

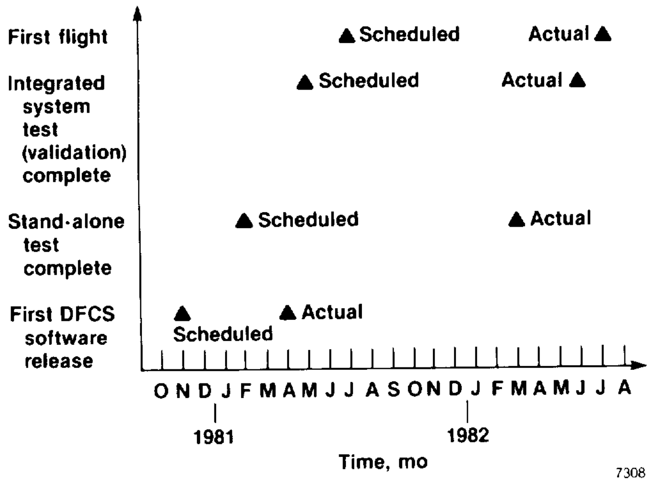


Figure 32. Qualification schedule.

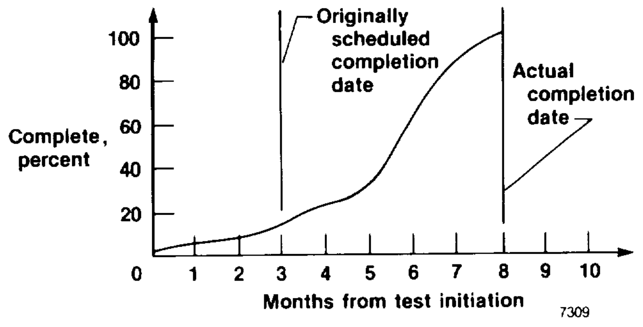


Figure 33. History of software coding and testing.

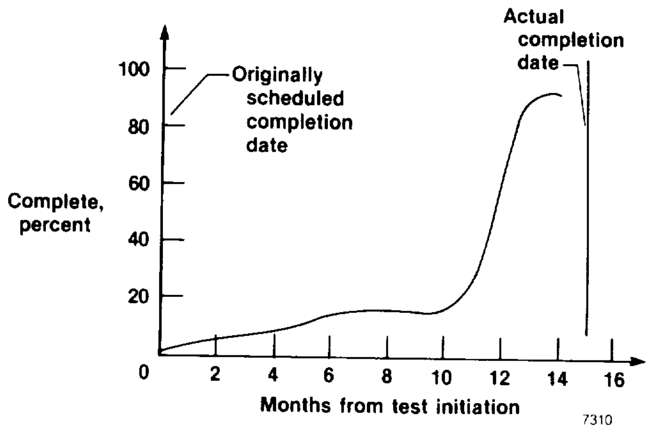


Figure 34. History of software verification.

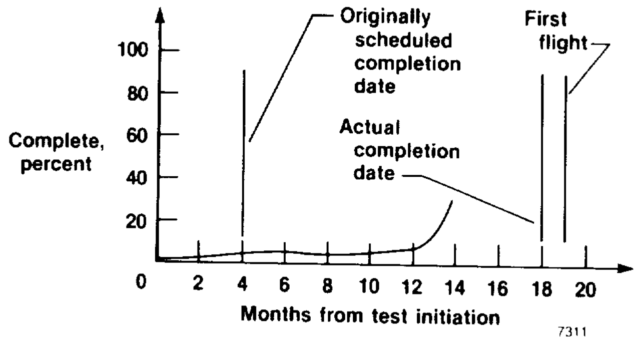
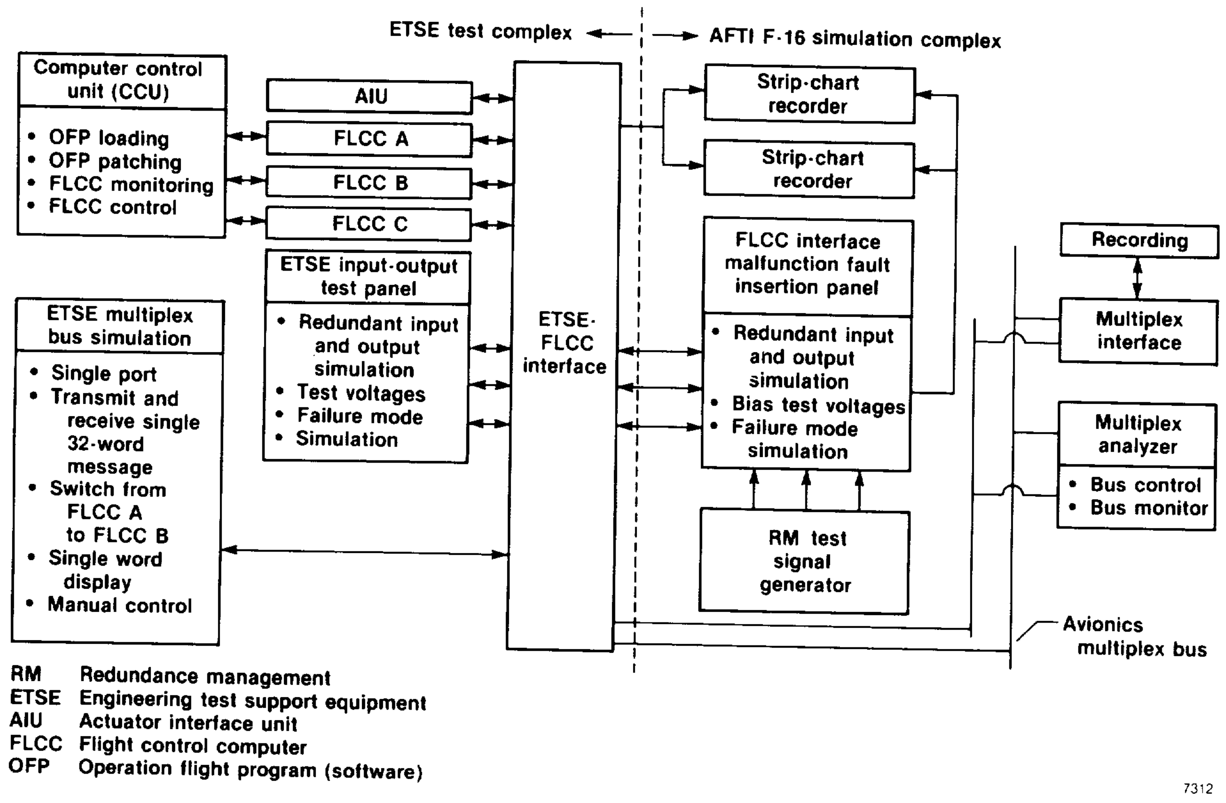


Figure 35. History of system validation.



RM	Redundance management
ETSE	Engineering test support equipment
AIU	Actuator interface unit
FLCC	Flight control computer
OFP	Operation flight program (software)

Figure 36. Support equipment for verification testing.

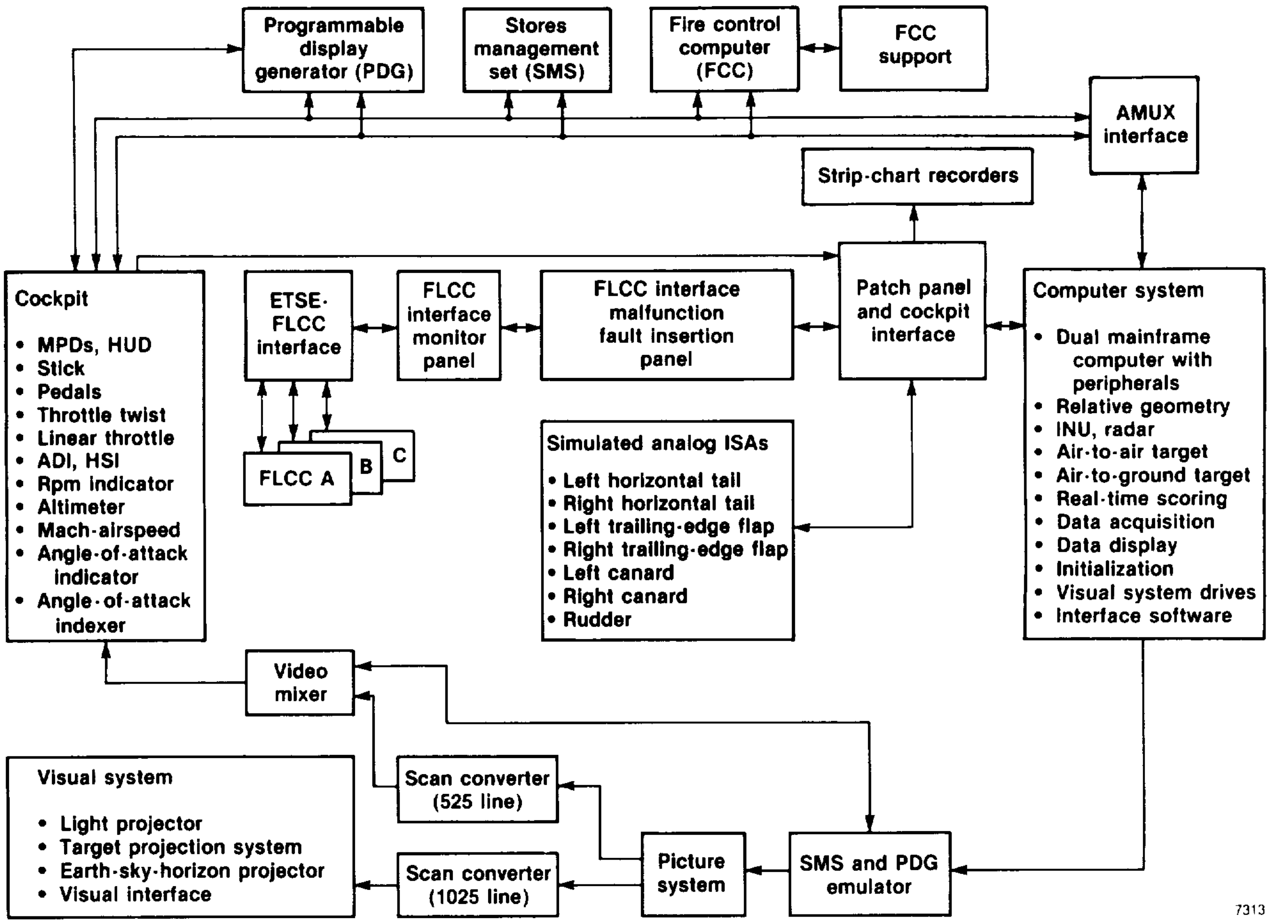


Figure 37. Validation support equipment.

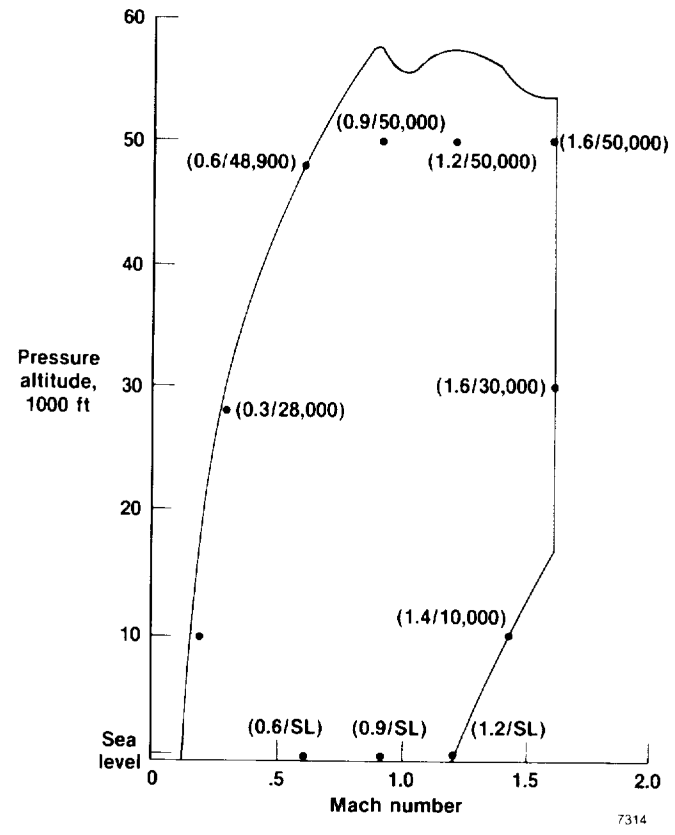


Figure 38. Phase 1 stress test conditions.

Point A Mach 1.2 at 500 ft, maximum g, 90° pitch up
Point B Perform desired maneuver from table
Point C Pitch over to −90° pitch attitude
Point D Perform desired maneuver from table
Point E 11,000 ft begin maximum g pullout
Point F Mach 1.2/5000 ft, reset
Note:
Throttle set at maximum afterburner power throughout test trajectory for phase 2

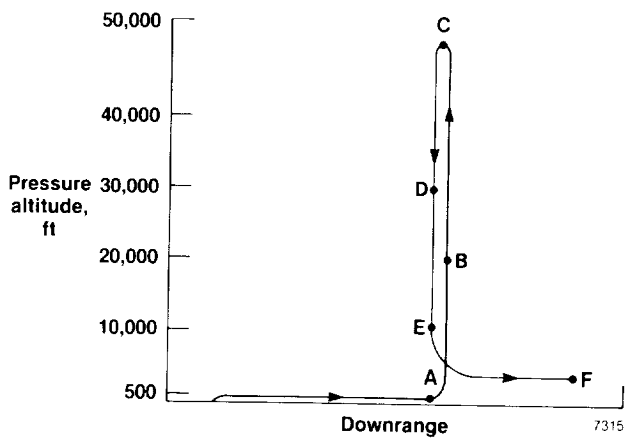


Figure 39. Stress test, inside loop.

Type	Second failure / First failure	None	Engine	EPU	Main generator	Hydraulic system A	ISA SV1 or SV 2	Hydraulic system B	ISA SV3	ISA solenoids	FLCC power supply	CPU	IOC	A-D converter	26 V ac 800 Hz power	Sensors-controllers	External +15 V dc power	D-A converter	ISA primary coil	ISA secondary coil	LEF command	IOC input discretes	System input discretes	Wraparound input discretes	IOC output discretes	ISA output discretes	Fail annunciation discretes	BIT discretes	Data link transmitter	Data link receiver	AMUX	IBU	LEF command servo
1	Engine																																
2	EPU																																
3	Main generator																																
4	Hydraulic system A										X									X								X	X	X	X		
5	ISA SV1 or SV2																																
6	Hydraulic system B										X									X								X	X	X	X		
7	ISA SV3																																
8	ISA solenoids																																
9	FLCC power supply					X		X			X	X	X	X	X			X	X	X	X		X		X			X	X	X	X		
10	CPU											X																					
11	IOC												X																				
12	A-D converter													X				X											X	X			
13	26 V ac 800 Hz power														X														X	X			
14	Sensors-controllers													X		X																	
15	External ±15 V dc power																X																
16	D-A converter													X				X															
17	ISA primary coil																		X										X	X			
18	ISA secondary coil					X		X												X													
19	LEF command																				X												
20	IOC input discretes																					X											
21	System input discretes																						X										
22	Wraparound input discretes																																
23	IOC output discretes																								X								
24	ISA output discretes																																
25	Fail annunciation discretes																																
26	BIT output discretes					X		X			X								X									X					
27	Data link transmitter					X		X			X								X										X				
28	Data link receiver					X		X			X								X											X			
29	AMUX					X		X			X								X												X		
30	IBU																																
31	LEF command servo																																X

X - Planned test case ☐ *- Not Applicable*

7316

Figure 40. Failure modes and effects testing dual fail test matrix.

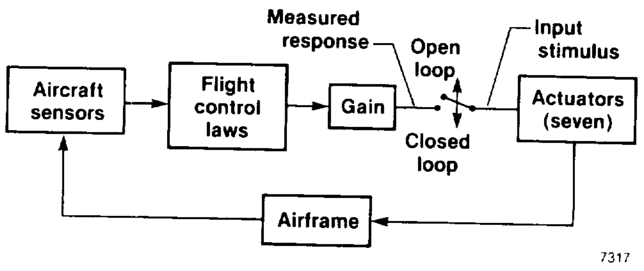


Figure 41. Setup for structural coupling tests.

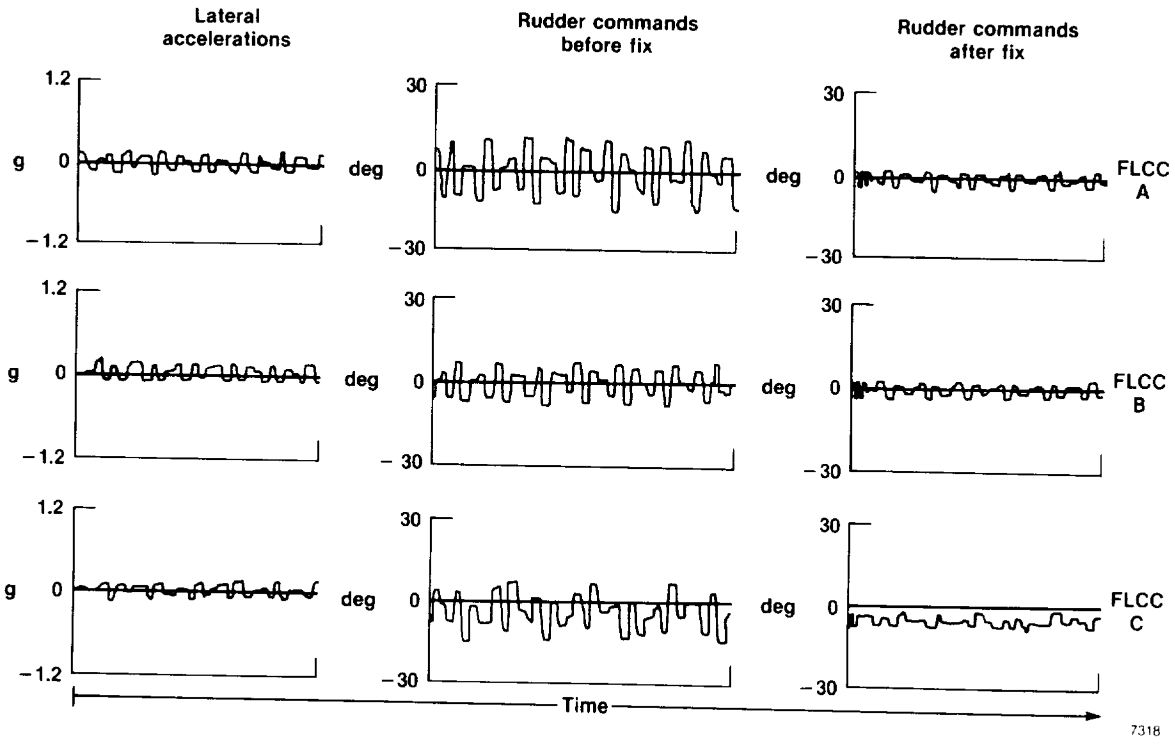


Figure 42. Gunfire test data.

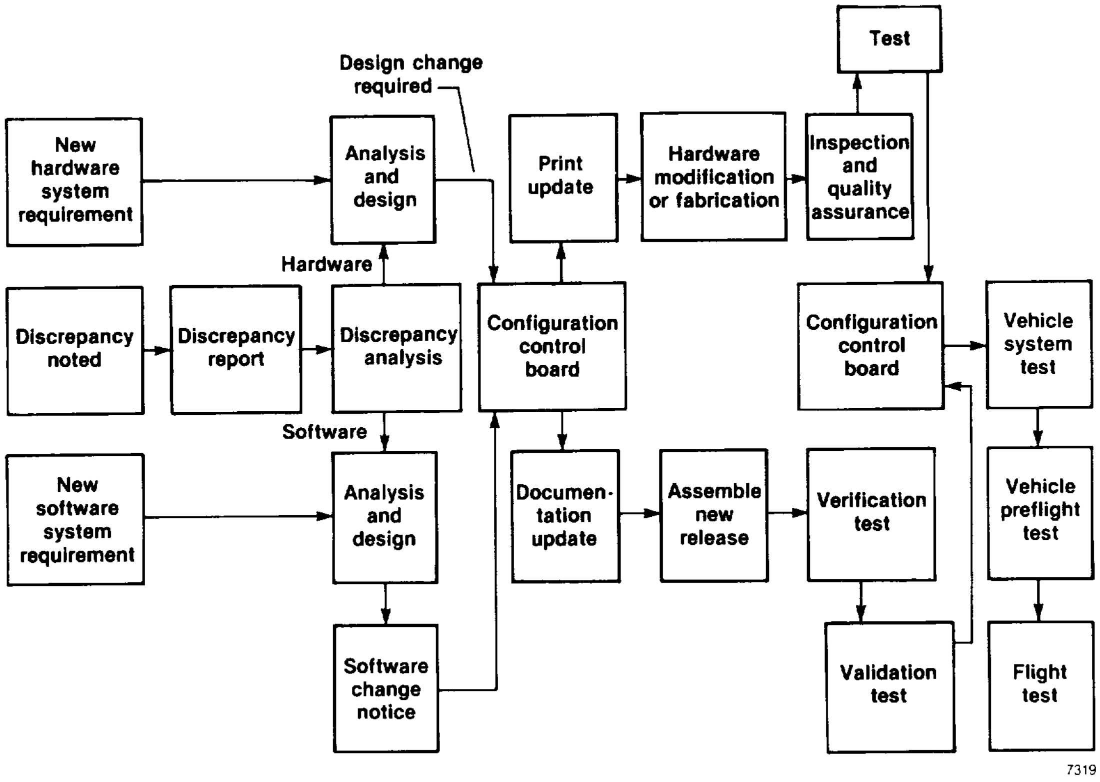


Figure 43. Configuration control process.

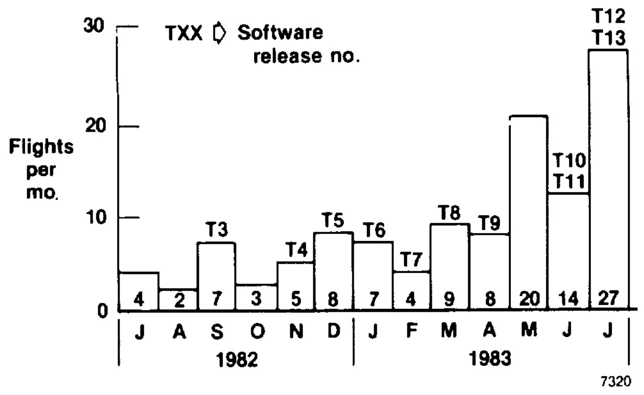


Figure 44. Flight test summary.

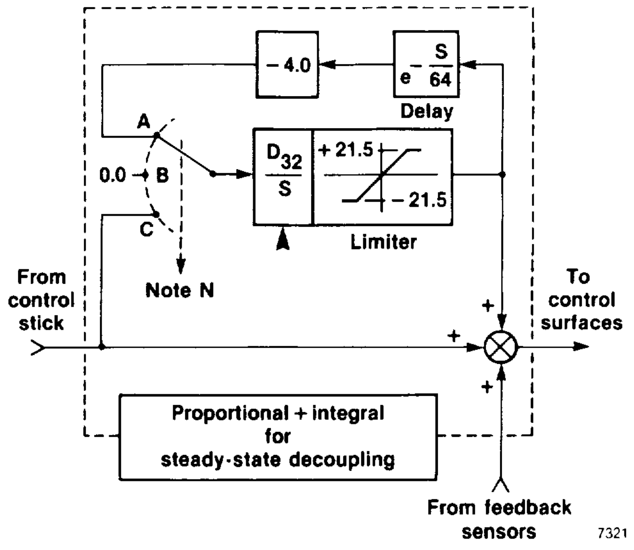


Figure 45. Example of a software switch used in the control laws.

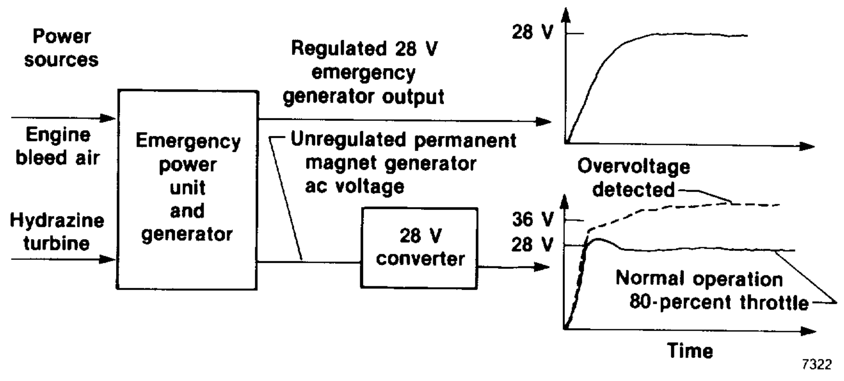

Power sources
Engine bleed air
Hydrazine turbine
Emergency power unit and generator
Regulated 28 V emergency generator output
Unregulated permanent magnet generator ac voltage
28 V converter
28 V
Overvoltage detected
36 V
28 V
Normal operation 80-percent throttle
Time
7322

Figure 46. Overview of electrical system over-voltage shutdown.

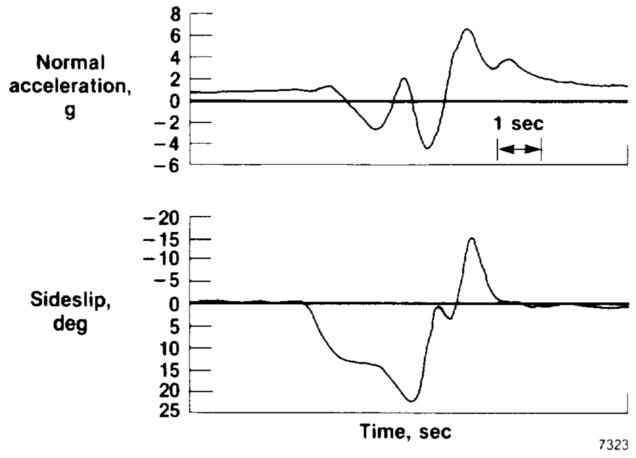


Figure 47. Normal acceleration and sideslip of flight 36 yaw departure.

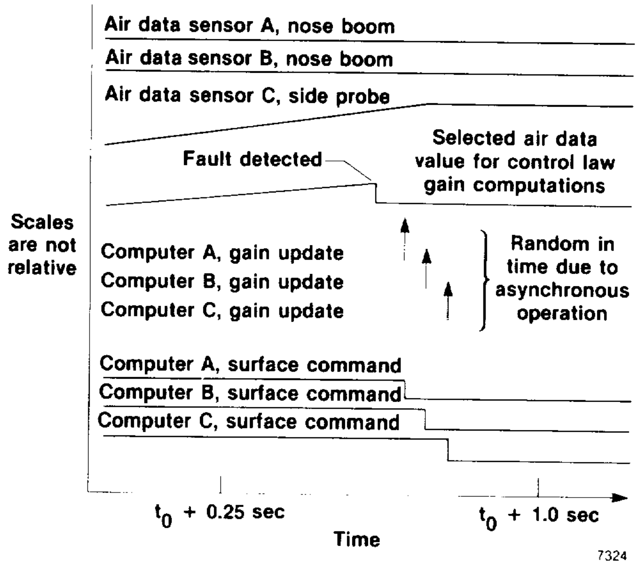

Air data sensor A, nose boom
Air data sensor B, nose boom
Air data sensor C, side probe
Fault detected
Selected air data value for control law gain computations
Scales are not relative
Computer A, gain update
Computer B, gain update
Computer C, gain update
Random in time due to asynchronous operation
Computer A, surface command
Computer B, surface command
Computer C, surface command
t_0 + 0.25 sec
t_0 + 1.0 sec
Time
7324

Figure 48. How an air data failure causes errors in surface commands.

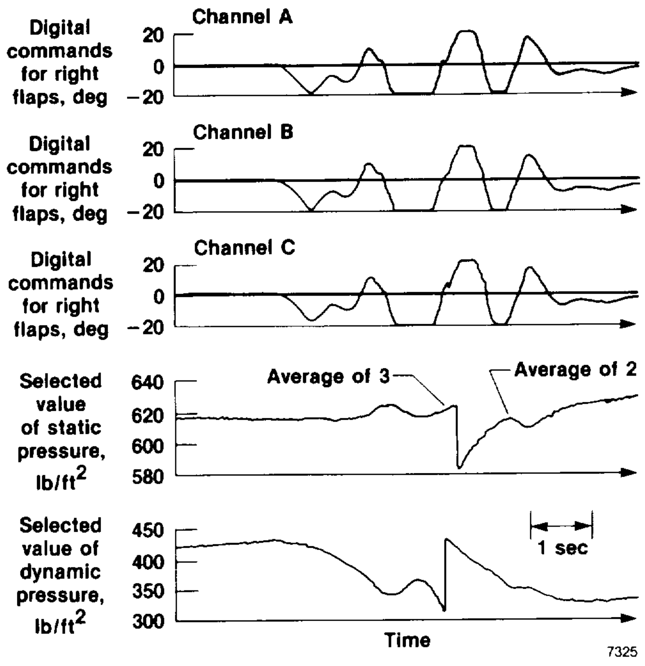


Figure 49. Air data transients and effects on surface commands.

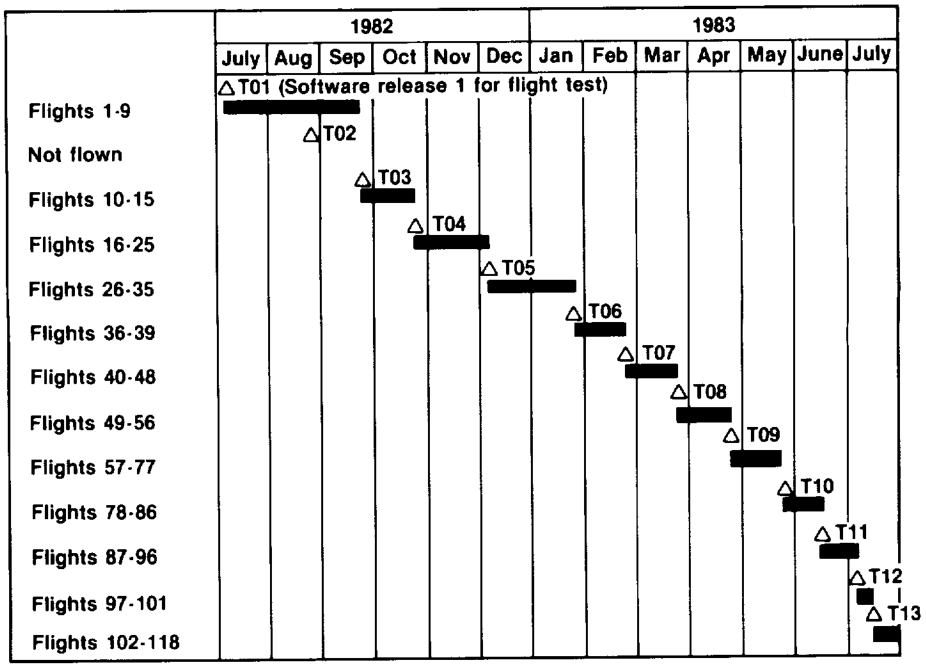


Figure 50. Software release summary.

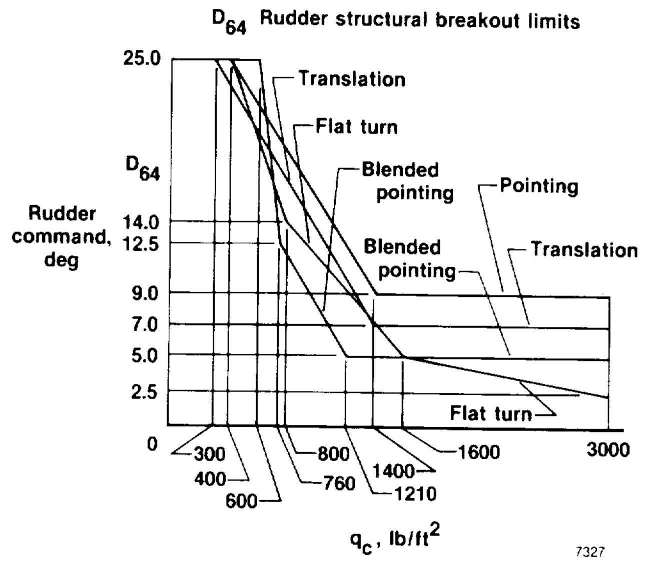
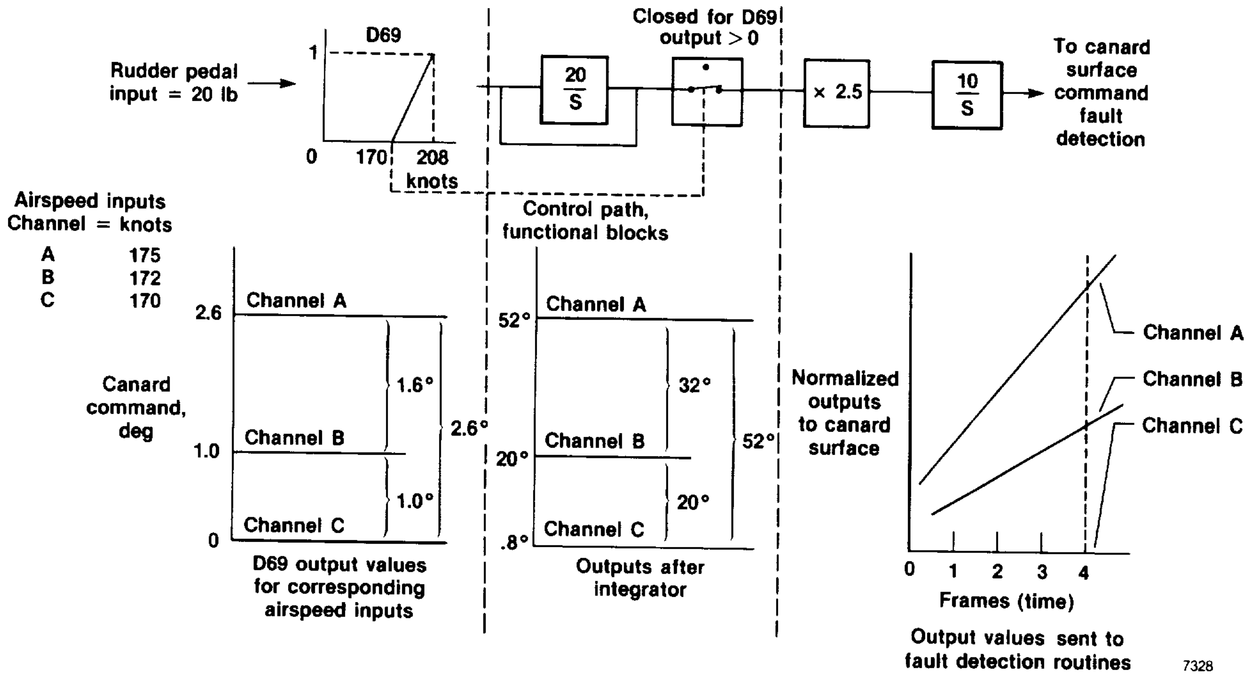


Figure 51. Rudder limit used to implement vertical tail load limiting.



Rudder pedal input = 20 lb
D69
1
0
170
208
knots
20/s
Closed for D69 output > 0
× 2.5
10/s
To canard surface command fault detection
Control path, functional blocks
Airspeed inputs Channel = knots
A 175
B 172
C 170
Canard command, deg
2.6
1.0
0
Channel A
Channel B
Channel C
1.6°
1.0°
2.6°
D69 output values for corresponding airspeed inputs
52°
20°
.8°
32°
20°
52°
Outputs after integrator
Normalized outputs to canard surface
Frames (time)
0 1 2 3 4
Output values sent to fault detection routines
7328

Figure 52. Summary of conditions causing flight 44 anomaly.

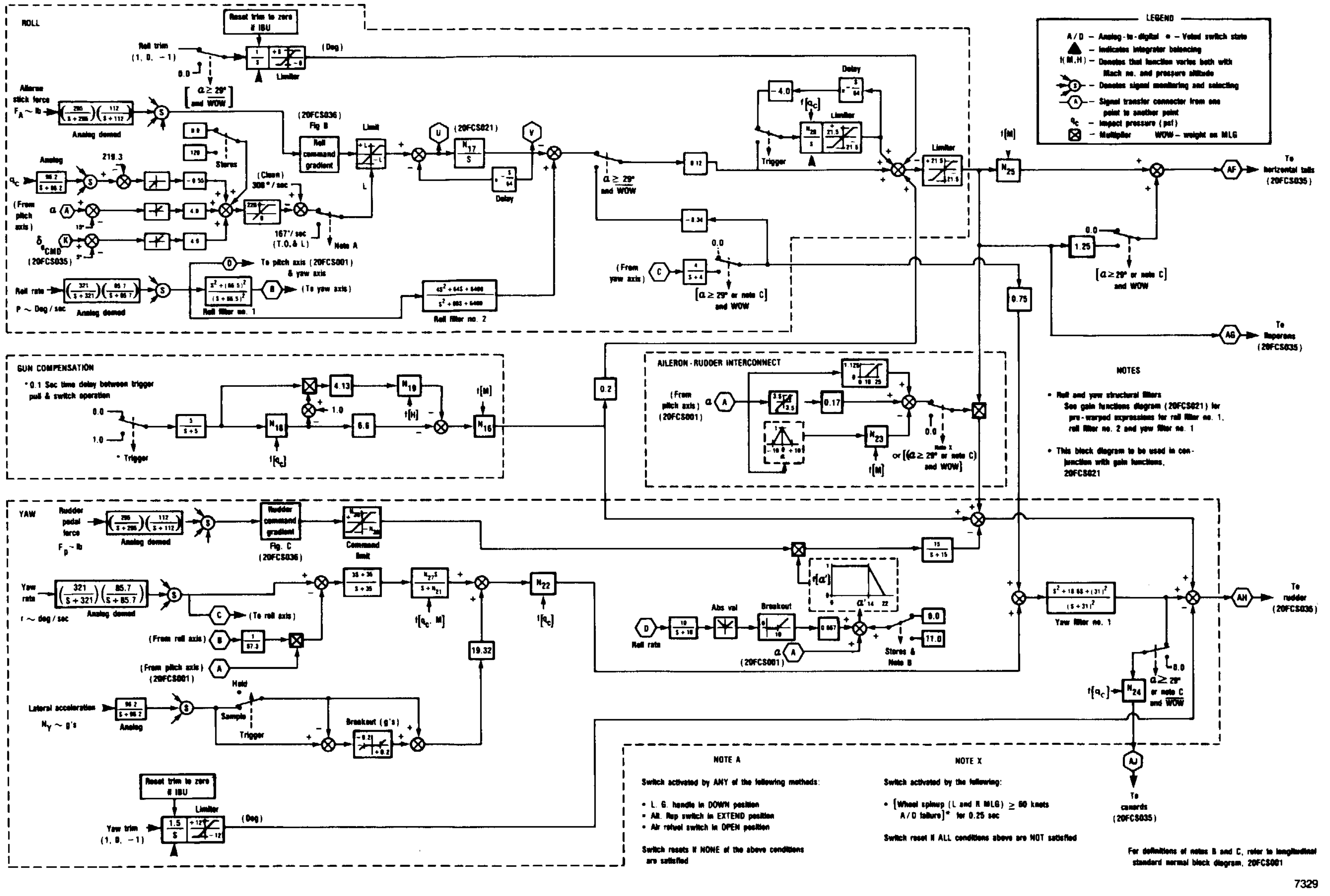
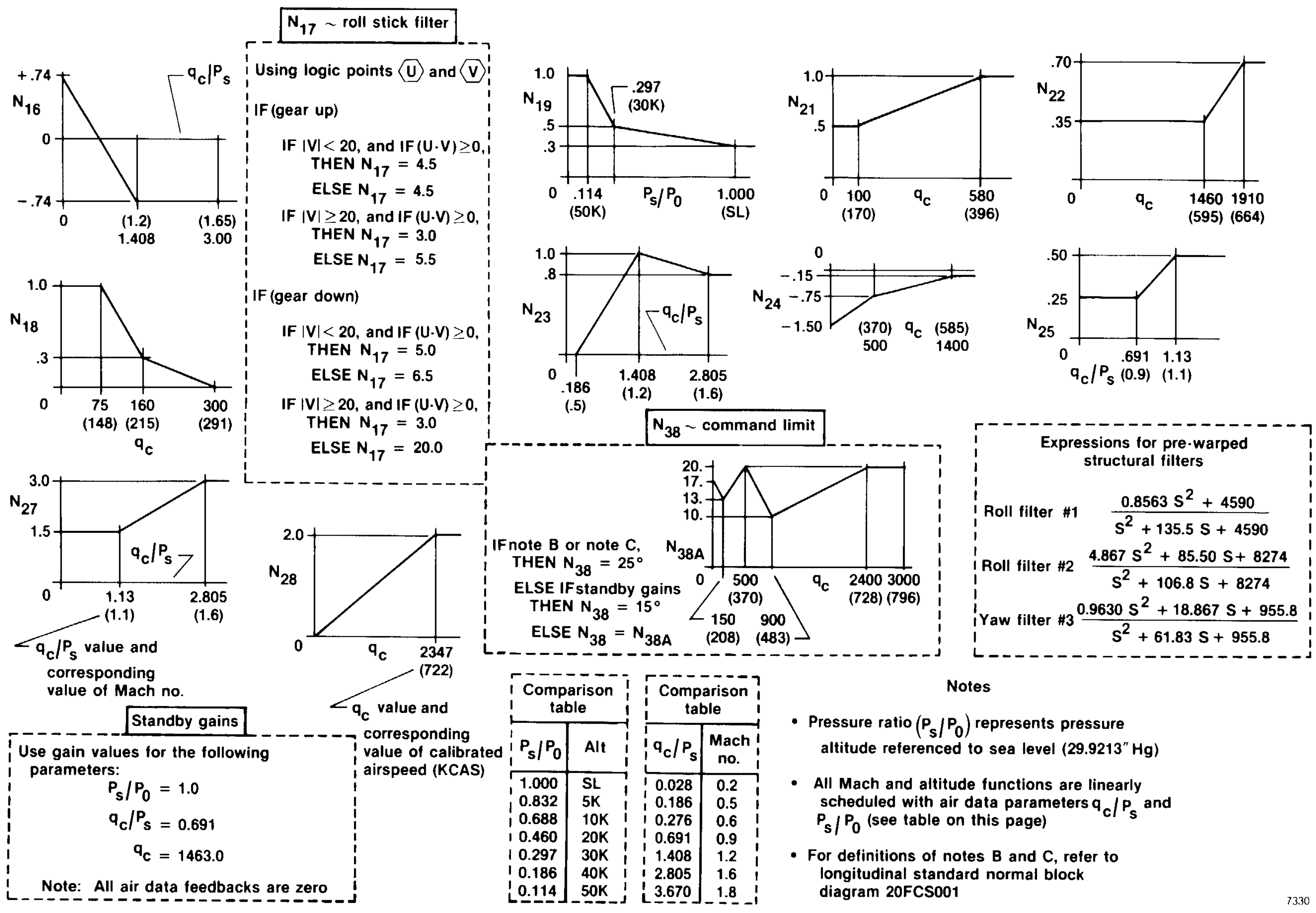

ROLL
Reset trim to zero @ IBU
Roll trim (1, 0, −1)
Limiter
(Deg)
Aileron stick force F_A ~ lb
Analog domed
(20FCS036) Fig B
Roll command gradient
Limit
(20FCS021)
Delay
$\alpha \geq 29°$ and WOW
Analog
q_c
(From pitch axis)
δ_{CMD} (20FCS035)
Stores
(Clean) 308°/sec
167°/sec (T.O. & L)
Note A
To pitch axis (20FCS001) & yaw axis
(To yaw axis)
Roll rate P ~ Deg/sec
Roll filter no. 1
Roll filter no. 2
(From yaw axis)
$[\alpha \geq 29°$ or note C$]$ and WOW
Trigger
GUN COMPENSATION
* 0.1 Sec time delay between trigger pull & switch operation
* Trigger
AILERON - RUDDER INTERCONNECT
(From pitch axis) (20FCS001)
or $[(\alpha \geq 29°$ or note C) and WOW$]$
Note X
YAW
Rudder pedal force F_p ~ lb
Rudder command gradient
Fig. C (20FCS036)
Command limit
Yaw rate r ~ deg/sec
(To roll axis)
(From roll axis)
(From pitch axis) (20FCS001)
Lateral acceleration N_Y ~ g's
Hold
Sample
Trigger
Breakout (g's)
Yaw trim (1, 0, −1)
Roll rate
Abs val
Breakout
Stores & Note B
Yaw filter no. 1
To rudder (20FCS036)
To horizontal tails (20FCS035)
To flaperons (20FCS035)
To canards (20FCS035)
LEGEND
A/D — Analog-to-digital • — Voted switch state
▲ — Indicates integrator balancing
f(M,H) — Denotes that function varies both with Mach no. and pressure altitude
Ⓢ — Denotes signal monitoring and selecting
Ⓐ — Signal transfer connector from one point to another point
q_c — Impact pressure (psf)
⊠ — Multiplier WOW — weight on MLG
NOTES
• Roll and yaw structural filters See gain functions diagram (20FCS021) for pre-warped expressions for roll filter no. 1, roll filter no. 2 and yaw filter no. 1
• This block diagram to be used in conjunction with gain functions, 20FCS021
NOTE A
Switch activated by ANY of the following methods:
• L. G. handle in DOWN position
• Alt. flap switch in EXTEND position
• Air refuel switch in OPEN position
Switch resets if NONE of the above conditions are satisfied
NOTE X
Switch activated by the following:
• [Wheel spinup (L and R MLG) ≥ 60 knots A/D failure]* for 0.25 sec
Switch reset if ALL conditions above are NOT satisfied
For definitions of notes B and C, refer to longitudinal standard normal block diagram, 20FCS001
7329

Figure 53. Lateral-directional standard normal mode (Anderson and Frank, 1984).



N_{17} ~ roll stick filter

Using logic points ⟨U⟩ and ⟨V⟩

IF (gear up)

IF $|V| < 20$, and IF $(U \cdot V) \geq 0$,
THEN $N_{17} = 4.5$
ELSE $N_{17} = 4.5$
IF $|V| \geq 20$, and IF $(U \cdot V) \geq 0$,
THEN $N_{17} = 3.0$
ELSE $N_{17} = 5.5$

IF (gear down)

IF $|V| < 20$, and IF $(U \cdot V) \geq 0$,
THEN $N_{17} = 5.0$
ELSE $N_{17} = 6.5$
IF $|V| \geq 20$, and IF $(U \cdot V) \geq 0$,
THEN $N_{17} = 3.0$
ELSE $N_{17} = 20.0$

N_{38} ~ command limit

IF note B or note C,
THEN $N_{38} = 25°$
ELSE IF standby gains
THEN $N_{38} = 15°$
ELSE $N_{38} = N_{38A}$

Standby gains

Use gain values for the following parameters:

$P_s/P_0 = 1.0$

$q_c/P_s = 0.691$

$q_c = 1463.0$

Note: All air data feedbacks are zero

Comparison table

P_s/P_0	Alt
1.000	SL
0.832	5K
0.688	10K
0.460	20K
0.297	30K
0.186	40K
0.114	50K

Comparison table

q_c/P_s	Mach no.
0.028	0.2
0.186	0.5
0.276	0.6
0.691	0.9
1.408	1.2
2.805	1.6
3.670	1.8

Expressions for pre-warped structural filters

Roll filter #1 $\dfrac{0.8563\ S^2 + 4590}{S^2 + 135.5\ S + 4590}$

Roll filter #2 $\dfrac{4.867\ S^2 + 85.50\ S + 8274}{S^2 + 106.8\ S + 8274}$

Yaw filter #3 $\dfrac{0.9630\ S^2 + 18.867\ S + 955.8}{S^2 + 61.83\ S + 955.8}$

Notes

- Pressure ratio (P_s/P_0) represents pressure altitude referenced to sea level (29.9213" Hg)
- All Mach and altitude functions are linearly scheduled with air data parameters q_c/P_s and P_s/P_0 (see table on this page)
- For definitions of notes B and C, refer to longitudinal standard normal block diagram 20FCS001

7330

Figure 54. Lateral-directional standard normal mode gain functions (Anderson and Frank, 1984).

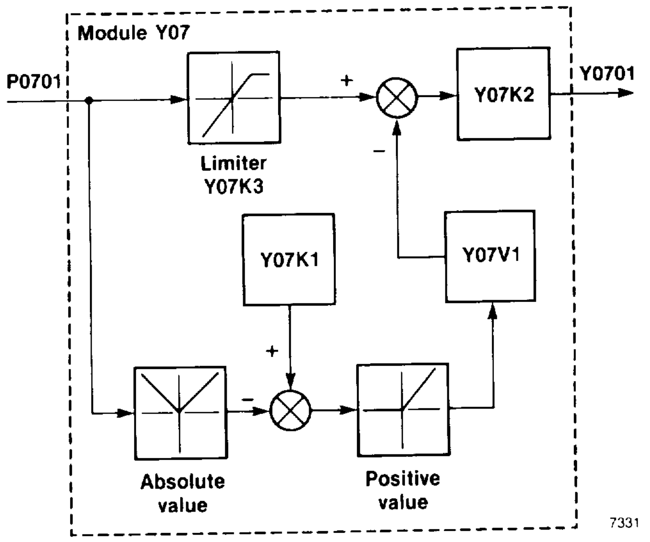


Figure 55. Angle-of-attack gain to aileron-rudder interconnect.

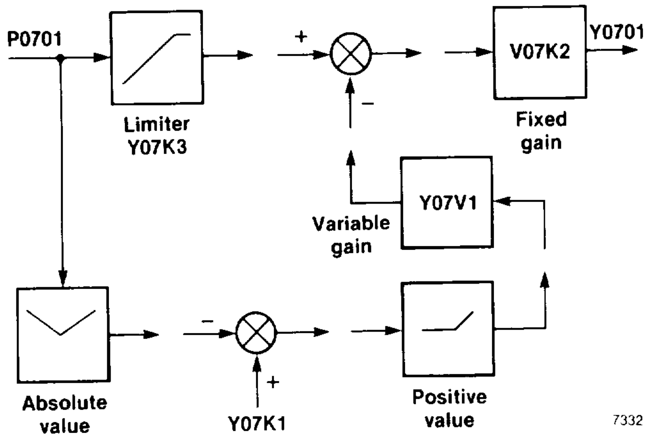


Figure 56. Control law module broken into individual blocks.

NASA National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TP-2857	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Development and Flight Test Experiences With a Flight-Crucial Digital Control System		5. Report Date November 1988
		6. Performing Organization Code
7. Author(s) Dale A. Mackall		8. Performing Organization Report No. H-1344
		10. Work Unit No. RTOP-999-12-08
9. Performing Organization Name and Address NASA Ames Research Center Dryden Flight Research Facility P.O. Box 273 Edwards, CA 93523-5000		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Technical Paper
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Engineers and scientists in the advanced fighter technology integration (AFTI) F-16 program investigated the integration of emerging technologies into an advanced fighter aircraft. AFTI's three major technologies included (1) flight-crucial digital control, (2) decoupled aircraft flight control, and (3) integration of avionics, flight control, and pilot displays. In addition to investigating improvements in fighter performance, researchers studied the generic problems confronting the designers of highly integrated flight-crucial digital control. The author provides an overview of both the advantages and problems of integration digital control systems. An examination of the specification, design, qualification, and flight test life-cycle phase is provided. An overview is given of the fault-tolerant design, multimoded decoupled flight control laws, and integrated avionics design. The approach to qualifying the software and system designs is discussed, and the effects of design choices on system qualification are highlighted.

17. Key Words (Suggested by Author(s)) Flight control Flight test Validation Verification	18. Distribution Statement Unclassified — Unlimited Subject category 08

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 116	22. Price A06

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161.

NASA-Langley, 1988